

Schlussbericht: Optimierte Kreislaufführung von Altglas durch digitale Füllstandsmessung

Themenoffene Projektförderung Deutsche Bundesstiftung Umwelt

Az. 37523/01-31

Zolitron - The Internet of Things Company GmbH
Universitätsstr. 136
44799 Bochum
Tel.: 0234/64082150
eMail: azinn@zolitron.com
Ansprechpartner: Dr.-Ing. Arndt-Hendrik Zinn

Bochum 2023

Inhaltsverzeichnis

1. Zusammenfassung	3
2. Einführung und Motivation	4
3. Vorgehensweise und Projektablauf	7
4. Projektergebnisse	10
5. Veröffentlichung	13
6. Fazit und Ausblick	14
7. Anhang: Bericht des Wuppertal Instituts	16

1. Zusammenfassung

Die Kreislaufwirtschaft ist eine der Schlüsselindustrien wenn es um die Bekämpfung des Fortschritts des Klimawandels geht. Durch Recycling kann der Einsatz von Primärrohstoffen reduziert werden, so dass diese Industrie am Ende nicht nur klimaneutral werden kann, sondern sogar Emissionen in anderen Sektoren reduziert, also bilanziell negativ sein kann. Allerdings erzeugt die Abfallsammlung als Teil der Prozesskette Emissionen, die besonders in dicht besiedelten, urbanen Gebieten produziert werden, wo der Abfall entsteht.

Die Abfallsammlung erfolgt in Deutschland noch anhand statischer Routen oder des subjektiven Gefühls der Leerungsfahrer, ohne die Vorteile der Digitalisierung bisher nutzen zu können. So wird vielfach zu häufig geleert, also Luft bewegt, oder es entsteht bei dem Versuch einer Reduzierung der Leerungsfrequenz eine Überfüllung der Container und so unbeabsichtigte Beistellungen. Die bisher verfügbaren Sensoren zur Digitalisierung der Behälter, die dieses Problem lösen könnten, waren zu teuer und aufwändig in der Installation und Wartung, so dass eine sensorbasierte Routenoptimierung nicht flächendeckend wirtschaftlich war.

Als Hochschulausgründung der Ruhr-Universität Bochum hat Zolitron ein revolutionäres Messverfahren entwickelt, welches basierend auf künstlich intelligenten Algorithmen (KI) den Füllstand kostengünstig messen kann und so Wartungs- und Installationskosten dramatisch reduziert werden können, was den Einsatz der Technik wirtschaftlich macht. Prototypisch wurde dies an Glascontainern gezeigt und soll nun in diesem Projekt am Beispiel eines kommunalen Entsorgers in den produktiven Betrieb überführt werden. Hierzu wurde eine Routenplanungssoftware entwickelt und füllstands-basierte, optimierte Routen errechnet.

Die Entsorger sollen durch die ökonomischen Vorteile intrinsisch motiviert werden, die ökologischen Einsparungen, die sich aus der Reduzierung der Leerungsfahrten ergeben, zu heben. Das Wuppertal Institut hat die Arbeiten begleitet und die ökologischen und ökonomischen Vorteile im Sinne eines Vorher-/Nachher-Vergleichs am Beispiel des kommunalen Entsorgers quantifiziert. Der Bericht wurde veröffentlicht und es konnte gezeigt werden, dass eine Einsparung von 25% Fahrstrecke und CO₂-Emissionen am Beispiel des Bochumer Entsorgungsunternehmens in der Glassammlung möglich sind. Nach Ablauf des Projekts wird die Routenoptimierung weiterentwickelt und auf weitere Arbeitsbereiche der Entsorgungswirtschaft angewandt. Es konnten bereits mit Abschluss des Projekts weitere Entsorger gefunden werden, die die Routenplanung künftig in ihrem täglichen Geschäft einsetzen werden und so Ihre Abfallsammlung, Straßenreinigung und Winterdienst optimieren.

2. Einführung und Motivation

Klimaschutz, Nachhaltigkeit, Ressourceneffizienz. Schlagworte die uns ständig begegnen und für die keine Branche so sehr steht wie die Kreislaufwirtschaft. Eine Multimilliarden-Euro Industrie in Deutschland mit zentralen Hubs in NRW, die für unsere heutige aber auch zukünftige Daseinsvorsorge essenziell ist. Auch international ist Deutschland mit 18% am Weltmarkt einer der größten Player der Kreislaufwirtschaft und für die Reduktion des Verbrauchs von Primärrohstoffen bestimmend.

Bei allem Lob findet sich allerdings auch eine Schattenseite, die Roland Berger 2016 in einer Studie aufgedeckt hat: Mit 32 von 100 möglichen Punkten ist die Kreislaufwirtschaft Schlusslicht in Deutschland wenn es darum geht die Vorteile der Digitalisierung zu nutzen. Seit Gründung der Zoliton im Jahr 2016 beobachten wir beim Großteil der Entsorger, Private und Kommunale, Betriebsabläufe wie im Jahr 1997: Fahrer bekommen händische Listen oder Excel-Tabellen zugesteckt, die sie nach ihrer Erfahrung abfahren, Routenplanung mit kleinen Fähnchen auf der großen Karte an der Wand, und die einzige digitale Infrastruktur ist das ERP-System zur Pflege der Kundendatenbank und der Buchhaltung. Oft kennen die Entsorger die KPIs ihrer eigenen Logistik nicht und auch der Datenaustausch mit den neu aufkommenden digitalen Wertstoffmärkten gestaltet sich schwierig. Dabei schlummert großes Optimierungspotenzial in Ihnen, sowohl um Emissionen und Kosten zu senken, aber auch um den Service für die Bevölkerung zu verbessern und die Recyclingquote zu erhöhen.

Aktuell existieren viele Teillösungen am Markt: Zunächst müssen die Assets der Entsorger - die Behälter - digitalisiert werden. Eine große Herausforderung, da Daten tausender, weit verstreuter Behälter zu minimalen Kosten über viele Jahre aggregiert werden müssen. Hersteller wie SSI Schäfer, Sensoneo, Nordsense, MOBA, Sloc und viele weitere bieten Ultraschallsensoren (Füllstandssensoren) an, die teuer sind, aufwendig im Innern der Behälter installiert werden müssen und deren Batterie nur ein bis max. zwei Jahre Energie liefert. Andere Sensoren liefern z.B. die GPS-Position der Container oder nutzen RFID um die Behälter eindeutig zu identifizieren, wenn die Leerungsfahrzeuge sie bewegen. Andere Hersteller wie MONOLOGA, InsideIT, Enwis, SAP und Weitere bieten Software an, um den Betriebsablauf möglichst passend digital abzubilden und zu verwalten, z.B. in Form einer Containermanagementsoftware. Wiederum andere Firmen bieten Routenoptimierung, Telematik für die Fahrzeuge oder Handheld-Geräte an. Eine sinnvolle Verbindung der einzelnen Lösungen ist Aufgabe jedes einzelnen Entsorgers; eine Aufgabe an der aktuell selbst Branchenriesen wie Remondis, PreZero und Alba zu knabbern haben, ganz zu schweigen von kleineren kommunalen Entsorgern meist ohne große IT-Abteilung. Insbesondere im Hinblick auf die Klimaschutzziele sowie die lokale und globale Verantwortung zur Reduktion von Treibhausgasemissionen zeigt sich nach bisherigem Erkenntnisstand für die Umstellung auf bedarfsorientierte Systeme ein enormes Potenzial der Kreislaufwirtschaft. Diese hat in den vergangenen 25 Jahren durch einen erfolgreichen Transformationsprozess jährlich ca. 30 Mio.

Mg CO₂-Äquivalente eingespart¹ und dürfte heute die einzige Wirtschaftsbranche sein, die durch ihre Geschäftstätigkeit mehr Treibhausgase vermeidet als sie erzeugt. Ansatzpunkte ergeben sich neben dem Bereich der Verwertungswege, also der ressourcenschonenden Rückfuhr von Abfällen in den Kreislauf, insbesondere auch bei der Steigerung der Effizienz im Bereich der Logistik der Abfallsammlung.

Die hier zu entwickelnde sensorgestützte Optimierung der Abfallsammlung soll das vorhandene ökologische Optimierungspotenzial heben, indem die Entsorger durch die ökonomischen Vorteile intrinsisch motiviert werden. Große private Entsorger wie Remondis rechnen mit einem Optimierungspotenzial von ca. 20% ihrer Logistikprozesse.

Am Beispiel der Glascontainer haben wir in den ersten 4 Monaten des Jahres 2021 bei 1.200 Altglascontainern 8.456 Leerungen erfasst. Dabei wurde ein durchschnittlicher Füllstand vor Leerung von nur 59,8% gemessen. Theoretisch waren also 1.775 Leerungsfahrten mit schweren Spezial-LKWs in dicht besiedelten Gebieten überflüssig. Das entspricht in diesem begrenzten Ausschnitt unserer Feasibility Study an Altglascontainern einer Einsparung von 10.650 LKW-Kilometern, 4.260 Litern Diesel und somit 4,51t CO₂.

Fraglich ist bisher, wie es bei anderen Abfallfraktionen aussieht und welcher Teil davon durch Digitalisierung der Betriebsabläufe gehoben werden kann, um eine Digitale Dividende zu zahlen und so die Entsorger intrinsisch zu motivieren die Technik einzusetzen, um Emissionen zu reduzieren, Stadtsauberkeit zu verbessern und die Recyclingquote zu steigern.

Das Umweltbundesamt (UBA) veröffentlicht jährlich Zahlen zum Abfallaufkommen in Deutschland und weißt für 2018 ein Aufkommen von Siedlungsabfällen i.H.v. 50,2 Mio. Tonnen aus, wobei dieses Niveau in den letzten Jahren gehalten wird. Eine Studie für das UBA aus 2010 weist eine durchschnittl. Belastung des Klimas durch Sammlung und Transport von Siedlungsabfällen von 40,2 kg CO₂-Äq/t aus. Bei dem Gesamtaufkommen an Siedlungsabfällen ergibt sich also eine Belastung von gut 2 Mio. Tonnen CO₂-Äquivalent und im entsprechenden Verhältnis auch 0,33 Mio. Tonnen NO_x und 0,12 Tonnen Feinstaub. Da die Sammlung fast ausschließlich mit konventionellen Diesel-LKWs in bevölkerungsreichen Gegenden geschieht, wo der Siedlungsabfall entsteht, werden diese schädlichen Emissionen vor allem in der Nähe des Menschen erzeugt.

Durch sensorgestützte Routenoptimierung wird demnach ein Rückgang der Emissionen um 20-46% erwartet. In diesem Projekt wurde ein ganzheitliches System zur füllstands-basierten Routenoptimierung entwickelt und die Optimierung am Beispiel eines großen kommunalen Entsorgers bewertet. Die gewonnenen Erkenntnisse sollen weitere Entsorger in Deutschland und zukünftig europaweit dazu motivieren, dieses Optimierungspotenzial zu nutzen, so dass ein Schneeballeffekt erwartet und nach Ablauf der Projektlaufzeit weitere Entsorger das System anfangen aktiv zu nutzen.

¹ BMU „Klimaschutz in Zahlen - Fakten, Trends und Impulse deutscher Klimapolitik“, Ausgabe 2020, S. 44
Zoltron - The Internet of Things Company | Universitätsstraße 136 | 44799 Bochum | t. +49 234 6408215-0 | e. info@zoltron.de |

Wir sind der Überzeugung, dass sich im Laufe dieses Jahrzehnts die dyn. Routenplanung auf Basis von Sensorinformationen flächendeckend in der Kreislaufwirtschaft etablieren wird. Mit diesem Projekt wollen wir den Übergang dazu wissenschaftlich begleiten und so ein Teil dazu beitragen. Es soll gezeigt werden, dass eine Einsparung der Fahrstrecke und CO₂-Emission von 20 - 45% möglich ist und so die Entsorger über die ökonomischen Vorteile motiviert werden die ökologischen Verbesserungen zu realisieren.

3. Vorgehensweise und Projektablauf

Die letztendliche Zielsetzung dieses Projekts ist die wissenschaftliche Beantwortung der Frage: **Wieviel Optimierungspotenzial steckt in kommunalen und privaten Entsorgern, wieviel davon kann durch Digitalisierung der Betriebsabläufe gehoben werden und wieviel Emissionen können dadurch reduziert werden?** Ebenfalls gilt es zu beantworten, wie sekundäre Effekte genutzt werden, z.B. zur Verbesserung der Stadtsauberkeit. Diese Fragen wurden bisher noch nicht beantwortet. Große private Entsorger wie Remondis rechnen mit einem Optimierungspotenzial von ca. 20% ihrer Logistikprozesse. Am Beispiel der Glascontainer haben wir in den vergangenen Monaten über alle Z-Nodes einen durchschnittlichen Füllstand vor Leerung von 46% gemessen, also ein rechnerisches Optimierungspotenzial von 54%. Fraglich ist bisher, welcher Teil davon durch Digitalisierung der Betriebsabläufe gehoben werden kann um eine Digitale Dividende zu zahlen und die Entsorger so intrinsisch zu motivieren die Technik einzusetzen, um Emissionen zu reduzieren, Stadtsauberkeit zu verbessern und die Recyclingquote zu steigern. Hierzu sollte im Projekt ein Vorher/Nachher-Vergleich der Optimierung durchgeführt werden.

Um den Vorher/Nachher-Vergleich zu bewerkstelligen, wurde mit den assoziierten Partner USB Bochum (kurz: USB) als innovatives Beispiel eines kommunalen Entsorgers im urbanen Gebiet zu Beginn des Projekts der Ist-Zustand gemessen. Der USB hatte vor Projektbeginn bereits alle Glascontainer mit der innovativen Füllstandssensorik von Zolitron ausgestattet, so dass zu Beginn des Projekts bereits wichtige logistische KPIs wie der Füllstand vor Leerung gemessen wurden, die Tonnage pro Kilometer der LKWs und der Grad an Überfüllung der Container. Weitere KPIs, die für den Vergleich betrachtet werden sollten, war die gesammelte Glasmenge und die gefahrenen Kilometer. Beide Werte sind leicht zu ermitteln, da der USB die Wiegezettel der Glas-LKWs theoretisch bereits seit Jahren mittels eigener Software in den LKWs protokolliert und auch den Kilometerstand der LKWs jeden Tag dokumentiert. Beides fand in der Praxis jedoch nicht statt, so dass ein Mitarbeiter des USB händisch die Wiegezettel und die Abfahrtskontrollprotokolle der LKWs digitalisieren musste.

Zunächst wurde der Zeitraum zum Projektstart Q1&2/2022 ausgewertet, um den Ist-Zustand vor der Optimierung zu erfassen. Es ist aufgefallen, dass wenige Leerungen pro Tag gefahren wurden und viele Container übervoll geleert wurden. In 2022 hat es personelle Änderungen im Fahrpersonal und der Disposition gegeben, so dass sich das Leerungsverhalten über den Projektzeitraum hinweg geändert hat. Um den dann neuen Ist-Zustand vor der Optimierung als Vergleichsgrundlage zu erfassen wurde ebenfalls Q1&2/23 ausgewertet. Die Ergebnisse finden sich im Kapitel 4.

Die Aufgabe bei Zolitrion während des Projekts lag bei der Entwicklung der Routenoptimierung und so letztendlich der Erstellung von Routen nicht nur für den USB, sondern die Software so zu entwickeln dass sie skalierbar auf viele Entsorger mit unterschiedlichen Routenanforderungen angewandt werden kann. Dazu wurden folgende Komponenten entwickelt:

- Kalibrierung der Füllstandsmessung um genaue Mengenangaben der Glascontainer zu erhalten.
- Implementation einer Routenoptimierung in die Z-Cloud. Die Optimierung soll in drei Schritten erfolgen:
 1. Nutzung der gewohnten Routen, wobei aber wenig befüllte Container nicht mehr angefahren werden.
 2. Neue dynamische Routen werden auf Basis der Füllstände tagesaktuell erzeugt.
 3. Stark genutzte Containerstellplätze werden mit weiteren Containern ausgestattet, um das benötigte Leerungsintervall der Bezirke zu homogenisieren und so die neuen, dynamischen Routen weiter zu optimieren.

Die Routenoptimierung ist ein mathematisches Optimierungsproblem, welches zunächst von der Aditabyte bearbeitet wurde und im Rahmen dieser Entwicklung in die Z-Cloud implementiert werden sollte. Es stellte sich jedoch heraus, dass die Anforderungen der Entsorgerbranche nicht zufriedenstellend in die Routenoptimierung von Aditabyte implementiert werden konnte. Es wurde festgestellt, dass das Kartenmaterial in Deutschland von fünf großen Anbietern (Google, HERE, TomTom, Bing, Open Street Map) nicht gleich gut waren und teilweise Straßen nicht gepflegt und gerade im ländlichen Bereich fehlerhaft war. Ebenfalls unterstützen nicht alle Kartenanbieter LKW-Daten (z.B. Höhen- und Durchfahrtsbeschränkungen) oder liefern keine Verkehrsinformationen. Kein Anbieter liefert alle drei Features. Die Routenoptimierung muss ebenfalls Restriktionen wie max. Fahrtzeit, Beladung, Wendeverbot, Rückwärtsfahrverbot etc. berücksichtigen, was Aditabyte nicht konnte. So wurde beschlossen ein eigenes Routenoptimierungssystem auf Grundlage von Open Street Map zu entwickeln, was die besten LKW-Kartendaten bereitstellt und Änderungen schnell eingepflegt werden. Die benötigten Verkehrsdaten werden separat für jeden Kunden gekauft und fließen im Optimierungsschritt in die Routenberechnung mit ein. Für die Optimierung wurde auf Open Source Repositories wie Valhalla und MapLibre zurück gegriffen.

- Auf eine API zur Anbindung an die vorhandene IT-Infrastruktur wurde verzichtet, da die Entsorger Ihre eigenen Systeme nicht ausreichend schnell anpassen können. Stattdessen wurde eine eigene Routing-App für Android entwickelt, genannt Z-Guide, der die neuen Routen über eine Zolitrion-interne API erhält und sie speziell für jeden Fahrer als Turn-by-Turn Navigation zur Verfügung stellt. Der Z-Guide beinhaltet ebenfalls einen sog. Freeride-Modus, bei dem die Routen der Fahrer protokolliert werden, um so von der Erfahrung der Fahrer zu profitieren. Zukünftig sollen diese Informationen dem Machine

Learning zugute kommen um die optimierten Routen den in Realität gefahrenen Routen weiter anzupassen.

Im Q4/22 wurde dem USB erstmalig optimierte Routen zur Verfügung gestellt. Diese neuen Routen müssen im Detail besprochen werden, eine Umsetzung z.B. in Form einer Navigation auf dem Tablet im LKW muss durchgeführt werden und auch das Personal und die Personalvertretung beim USB muss geschult und informiert werden, um für eine hohe Akzeptanz der neuen Arbeitsweise zu werben. Durch personelle Engpässe beim USB konnte dieser Schritt allerdings nicht während der Projektlaufzeit durchgeführt werden, so dass entschieden wurde die neuen Routen zu simulieren und die Nachher-Werte für den Vergleich anhand dieser Simulationen auszurechnen.

Das Wuppertal Institut erhielt sowohl Vorher- als auch Nachher-Werte um eine wissenschaftlich fundiert, unabhängige Bewertung des CO₂-Einsparpotenzials zu erstellen, die die ökologische und ökonomische Sinnhaftigkeit der neuen Technik belegt. Mit der Veröffentlichung sollen so weitere Entsorger intrinsisch motiviert werden, das ökonomische Optimierungspotenzial zu heben, um damit verbunden die ökologischen Vorteile ebenfalls zu ernten.

4. Projektergebnisse

Die ausführlichen Ergebnisse sind im Bericht des Wuppertal Instituts nachzulesen. Das WI hat zunächst die CO₂-Emissionen für die Herstellung und den Betrieb der Füllstandssensoren berechnet. Diese Emissionen wurden im Verhältnis zu den eingesparten Emissionen mit 0,5% als vernachlässigbar gering angesehen. Der Fokus liegt also auf den durch die Optimierung eingesparten Emissionen. Im folgenden Histogramm wird der Ist-Zustand vor Optimierung ersichtlich:

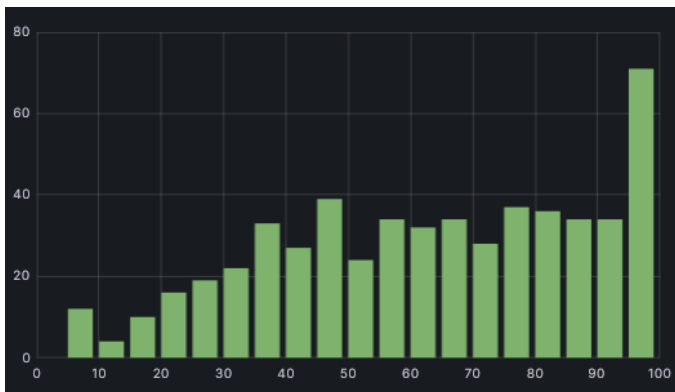


Abbildung 1: Histogramm des gemessenen Füllstands vor Leerung der Weißglascontainer beim USB im Juli 2023. Auf der X-Achse ist der Füllstand vor Leerung in Prozent aufgetragen. Auf der Y-Achse ist die gemessene Häufigkeit der Leerung eines Weißglascontainers bei dem jeweiligen Füllstand aufgetragen.

Im Betrachtungszeitraum Q1&2/23 wurden ca. 200km pro Werktag von zwei Fahrzeugen gefahren, wobei eines der Fahrzeuge kleiner ist als das andere Fahrzeug und nur ca. die halbe Kapazität des großen Fahrzeugs hat. Die Weißglascontainer, also die turnusführende Fraktion, wurden im Betrachtungszeitpunkt bei 61,1% durchschnittlichem Füllstand geleert. Wie in Abb. 1 zu erkennen ist wurden 13,5% der Leerungen bei >95% Füllstand durchgeführt wobei davon auszugehen ist, dass zu spät geleert wurde und es zu Beistellungen an den Containerstellplätzen kam. Der USB plant alle Leerungen in einem 4-wöchig wiederkehrenden Zyklus, wobei sich die Fahrer nicht zwingend an den Plan halten müssen. Im Betrachtungszeitraum wurden durchschnittlich 638 Leerungen in 4 Wochen gemessen. Es ist zu beachten, dass ein signifikanter Anteil der gefahrenen Tageskilometer auf die Fahrt vom Sammelgebiet in Bochum zur Kippstelle am Essener Hafen entfällt. Diese Wegstrecke kann nicht durch Optimierung reduziert werden. Es kann bei der Optimierung einzig darauf geachtet werden, dass der große LKW möglichst intensiv genutzt wird und der kleine LKW möglichst wenige Schichten übernehmen muss um die Stadtsauberkeit auf das gewünschte hohe Niveau zu bringen.

Die Optimierung wurde mit den beschriebenen Methoden durchgeführt. Da manche Container seltener als 4 Wochen zu leeren sind wurde ein 8-wöchiger Plan erstellt. Das folgende Histogramm zeigt die Simulation der Verteilung der Leerungsfüllstände nach der Optimierung.

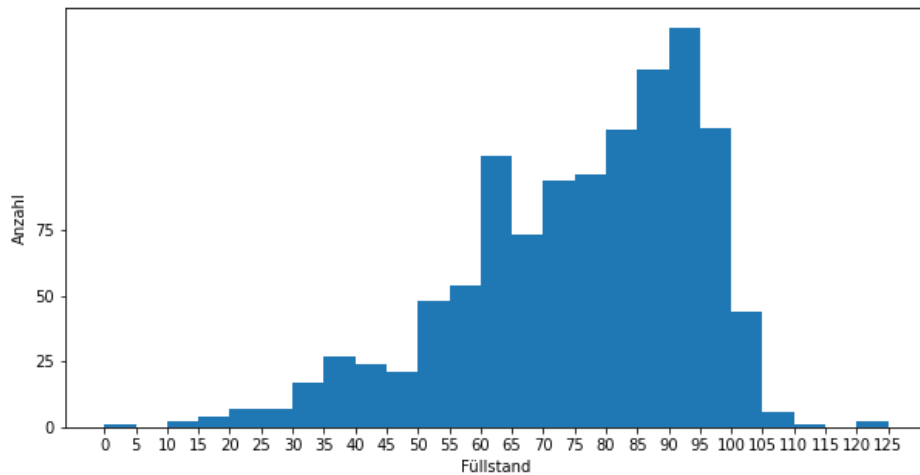


Abbildung 2: Histogramm des simulierten Füllstands vor Leerung der Weißglascontainer beim USB nach erfolgter Optimierung im 8-Wochen Plan. Auf der X-Achse ist der Füllstand vor Leerung in Prozent aufgetragen. Auf der Y-Achse ist die gemessene Häufigkeit der Leerung eines Weißglascontainers bei dem jeweiligen Füllstand aufgetragen.

Nach der Optimierung wurde ein durchschnittlicher Füllstand von 78% erreicht. Dieser Wert ist 17% höher als vor der Optimierung und ermöglicht eine bessere Auslastung der LKWs, so dass in 4 Wochen 587 Leerungen durchzuführen sind. Theoretisch wäre ein durchschnittlicher Füllstand vor Leerung zwischen 95-100% beabsichtigt, allerdings muss berücksichtigt werden, dass es vielmals effizienter ist einen Container einzusammeln der ggf. erst 60% voll ist aber dennoch mitgenommen wird, weil der LKW gerade in der Nähe ist und eine eigene Anfahrt dieses Containers wenige Tage später ineffizienter wäre. Die Zielgröße der Optimierung war die gefahrenen Kilometer und die Arbeitszeit zu senken, nicht alle Container zwingend voll abzuholen. Nach der Optimierung wurde eine tägliche Fahrtstrecke von 151,3km unter Berücksichtigung des Verkehrs simuliert, wobei der große LKW jeden Tag eingesetzt wird und der kleine LKW nur jeden zweiten Tag. Der zweite Fahrer ist an den übrigen frei und kann andere Fraktionen leeren und die Stadtsauberkeit verbessern. Die Stadtsauberkeit ist nach der Optimierung ebenfalls optimiert da neuralgische Containerstellplätze häufiger angefahren werden, so dass Container seltener überlaufen, vgl. Abb. 1 / 2.

Eine weitere Optimierungssiteration wurde durchgeführt unter der Annahme, dass weitere Container an besonders stark frequentierten Stellplätzen aufgestellt werden und so die Leerungsintervalle im Gebiet homogenisiert wird, was es dem Optimierungsalgorithmus einfacher macht. Das Ergebnis ist im folgenden Histogramm zu sehen:

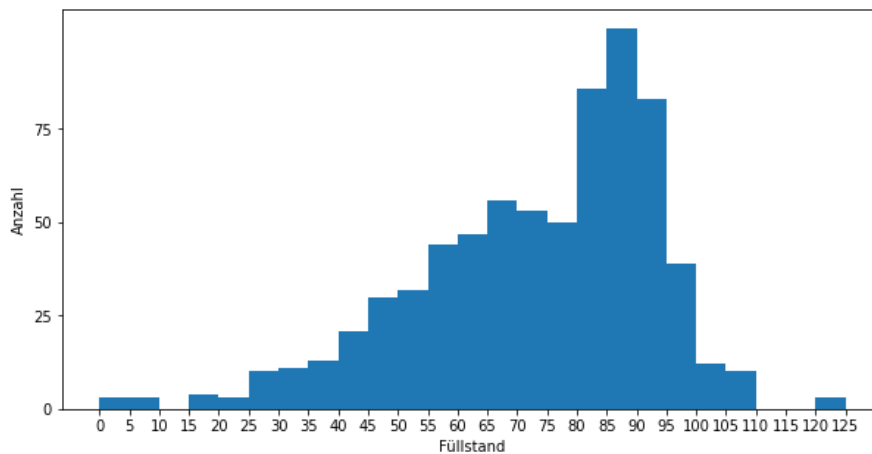


Abbildung 3 Histogramm des simulierten Füllstands vor Leerung der Weißglascontainer beim USB nach erfolgter Optimierung im 8-Wochen Plan und Anpassung der Container auf stark frequentierten Stellplätzen. Auf der X-Achse ist der Füllstand vor Leerung in Prozent aufgetragen. Auf der Y-Achse ist die gemessene Häufigkeit der Leerung eines Weißglascontainers bei dem jeweiligen Füllstand aufgetragen.

Hierbei konnte der durchschnittliche Füllstand vor Leerung auf 73,7% erhöht werden und die Fahrtstrecke auf 149,2km pro Werktag gesenkt werden. Weitere Optimierung scheinen möglich, es erschien uns allerdings zielführender zunächst diese Touren in der Realität zu testen, ehe die Stellschrauben weiter angezogen werden. Perspektivisch kann das Gebiet aber mit einem LKW geleert werden bei gleichzeitiger Verbesserung der Stadtsauberkeit gegenüber des Status Quo. Die folgende Tabelle aus dem Bericht des WI fasst die Ergebnisse zusammen:

	Status-Quo für Juli 2023	Optimierte Sammlung	Optimierte Sammlung mit Kapazitätsanpassung
Gesparte Strecke [km/Monat]	0	1275	1494
Gesparte Strecke [%]	0	21	25
Gesparte THG Emissionen [kg CO2eq]	-13,2	2775	3254
Gesparte THG Emissionen relativ zum Status Quo [%]	-0,1	21	25

Tabelle 1: Zusammenfassung der Ergebnisse der Optimierung. Vgl. Tabelle 3 Anlage

Das WI hat die Einsparung an CO₂-Emissionen auf Grundlage der verwendeten LKWs ausgerechnet und eine Reduktion des Ausstoßes nach der Optimierung um 25% ermittelt. Weiterhin führt das WI auch die Option an, Teile der Kosten des Betriebs der Sensorik über die gesunkenen CO₂-Emissionen und damit gesunkene CO₂-Bepreisung gegenzurechnen.

5. Veröffentlichung

Der Bericht des WI zum Projekt wurde im September 2023 veröffentlicht. Er wird in den nächsten Monaten über die Kanäle von Zolitron, Website, Social Media, Veröffentlichungen des VKU weiter verbreitet.

6. Fazit und Ausblick

Im Projekt wurde eine auf die Bedürfnisse und Anforderungen der Entsorgungswirtschaft ausgerichtete Routenoptimierung entwickelt. KPIs der Glassammlung wie gefahrene Strecke und geleerte Container wurden vor der Optimierung gesammelt. Aufgrund des personellen Mangels beim Entsorger konnte die Optimierung der Routen bis zum Abschluss des Projekts noch keine Anwendung finden. Die gleichen KPIs nach der Optimierung wurden simuliert und ausgerechnet. Der Vorher/Nachher-Vergleich hat am Beispiel der Glassammlung in Bochum ein Optimierungspotenzial von 25% Fahrstrecke, CO₂-Emissionen und Fahrzeit ergeben. Neben den Auswirkungen auf den Verkehr und die damit verbundenen Einsparungen in Kraftstoff und CO₂-Emissionen wurde während der Projektlaufzeit ebenfalls klar, dass die Entsorger teilweise noch besser zum Einsatz der Routenoptimierung zu motivieren sind über den eingesparten Personalaufwand bei gleichzeitig teils besserer Stadtsauberkeit, da die Branche immer stärker unter Personalmangel leidet aufgrund des Ausscheidens von älteren Fahrern und des hohen Krankenstandes. Zwar benötigt die Umstellung der Arbeitsweise und die Gewöhnung an die neuen Tools wie z.B. das Tablet mit der Routenführung anfänglich etwas Personaleinsatz, mittelfristig ergeben sich allerdings die gezeigten Optimierungen. Die neuen Tools erlauben es auch neues oder gebietsfremdes Personal kurzfristig produktiv einzusetzen, sollten die Stammfahrer ausfallen.

Zwar wurde die Routenoptimierung noch nicht in Bochum umgesetzt, allerdings wird die hier gestartete Routenoptimierung weiterentwickelt, verfeinert und für weitere Tätigkeitsbereich in der Abfallwirtschaft nutzbar gemacht. So konnten bis zum Abschluss des Projekts weitere Kommunen gewonnen werden, die die Routenoptimierung nutzen werden, sei es in der Müllsammlung, der Straßenreinigung oder dem Winterdienst. Wir erwarten, dass sich die füllstands-basierte Routenoptimierung mittelfristig zum Umsatztreiber der Zolitron entwickelt. So können wir aufgrund der oben beschriebenen ökonomischen Effekte (Personalaufwand & -mangel, Kraftstoff- & Fahrzeugkosten) die Entsorger motivieren die ökologischen Vorteile der Routenoptimierung zu realisieren und die vorhandenen Ineffizienzen zu reduzieren, was einem Rückgang von LKW-Verkehr und Emissionen gerade im dicht besiedelten Bereich entspricht.

7. Anhang: Bericht des Wuppertal Instituts

Bericht | September 2023

Ganzheitliche Kreislaufwirtschaft durch das Green IOT

—

Herausgeberin:

Wuppertal Institut für Klima, Umwelt, Energie gGmbH
Döppersberg 19
42103 Wuppertal
www.wupperinst.org

Autoren:

Dr. Phillip Bendix, Jonathan Klees, Katja Geiter
E-Mail: phillip.bendix@wupperinst.org

Bitte die Publikation folgendermaßen zitieren:

Bendix, P., Klees, J., Geiter, K. (2023). Ganzheitliche Kreislaufwirtschaft durch das Green IOT. Wuppertal Institut.

Inhaltsverzeichnis

Inhaltsverzeichnis	3
Verzeichnis von Abkürzungen, Tabellen und Abbildungen	4
1 Einleitung	6
2 Methodik	7
2.1 Datenauswertung	7
2.2 Routenoptimierung	9
2.3 Berechnung der CO ₂ Emissionen	9
2.3.1 <i>CO₂ Emissionen durch die Herstellung und den Betrieb des Sensors</i>	10
3 Ergebnisse und Diskussion	12
3.1 Daten zum Status Quo	12
3.1.1 <i>Glasmenge</i>	12
3.1.2 <i>Füllstandserfassung</i>	14
3.1.3 <i>Strecke der Sammelfahrzeuge</i>	17
3.2 Berechnung der CO ₂ Emissionen	19
3.2.1 <i>Emissionen durch die Herstellung der Sensoren</i>	20
3.2.2 <i>Emissionen durch den Betrieb der Sensoren</i>	20
3.2.3 <i>Emissionsausgleich durch kürzere Strecken der Sammelfahrzeuge</i>	21
3.3 Routenoptimierung	21
3.3.1 <i>CO₂ Einsparungen durch die Routenoptimierung</i>	24
4 Zusammenfassung und Fazit	26
5 Literaturverzeichnis	27
6 Anhang	28
6.1 Ausführliche Berechnung der Emissionen für den Betrieb der Sensoren	28
6.2 Jupiter Notebook und Source Code für die Datenauswertung	30

Verzeichnis von Abkürzungen, Tabellen und Abbildungen

Abkürzungsverzeichnis

Abkürzung	Bedeutung
IT	Informationstechnologie
USB	USB Bochum GmbH, Abfallentsorgungsbetrieb der Stadt Bochum
CO ₂	Kohlenstoffdioxid
CO ₂ eq	Kohlenstoffdioxid Äquivalente
%	Prozent, Faktor 0,01
s	Zeit in Sekunden
t	Gewicht in Tonnen, 10 ⁶ g
m	Strecke in Meter
d	Zeit in Tagen, 86400 s
MB	Speicher in Megabyte, 2 ²⁰ Byte
μW	Leistung in Mikrowatt, 10 ⁻⁶ W, bzw. 10 ⁻⁶ kg·m ² ·s ⁻³
W	Leistung in Watt, bzw. kg·m ² ·s ⁻³
g	Gewicht in g
kg	Gewicht in Kilogramm, 10 ³ g
km	Strecke in Kilometer, 10 ³ m
GB	Speicher in Gigabyte, 2 ³⁰ Byte
IOT	Internet of things, englisch für: Internet der Dinge

Tabellenverzeichnis

Tab. 3-1	Tabelle zur Verteilung der Füllstände bei Leerung der Container	15
----------	---	----

Abbildungsverzeichnis

Abbildung 1	Auszug aus dem Datensatz zur Erfassung der Glasmengen.	7
Abbildung 2	Auszug aus dem Datensatz zur Erfassung der Glasmengen.	8
Abbildung 3	Auszug aus dem Datensatz zur Erfassung der Strecken.	8
Abbildung 4	Beispiel für die Auswertung der Daten, hier Erstellen eines Diagramms zur Abbildung der Glasmengen mit der Glasmenge auf der Y-Achse, der Zeit auf der X-Achse sowie farblicher Codierung der Datenpunkte zur Unterscheidung von Braun-, Grün- und Weißglas .	9
Abbildung 5:	Aufbau Füllstandsensoren mit Deckplatte, Solarzelle, Gehäuseoberseite, Dichtung, Platine, Akku, Gehäuseunterseite und Schrauben (Abbildung: Zolitrion).	10
Abbildung 6	Visualisierung der erfassten Glasmenge.	12
Abbildung 7	Histogramm der Glasmengen zur Veranschaulichung der verschiedenen Maxima und Breiten der jeweiligen Verteilung.	13
Abbildung 8	Kumulierte tägliche Glasmenge und Anzahl der Leerungen für beide Fahrzeuge. 14	
Abbildung 9	Verteilung der Füllstände bei der Leerung der Container.	15
Abbildung 10	Füllstände für individuelle Container (Nummer 1 bis 20) bei Leerung für alle Container, die im Messzeitraum dreimal oder öfter geleert wurden.	17
Abbildung 11	Füllstand der Container zum Zeitpunkt der Leerung, aufgetragen gegen die Zeit. Die Füllstände der Container mit höchstem und niedrigstem Füllstand an diesem Tag sind jeweils als maximaler bzw. minimaler Füllstand eingezeichnet.	17
Abbildung 12	Zurückgelegte Strecke sowie gesammelte Glasmenge der beiden Sammelfahrzeuge aufgetragen gegen die Zeit.	19

- Abbildung 13 Ausgabe von openLCA zur Aufteilung der Treibhausgasemissionen auf die
verschiedenen vorgelagerten Prozesse der Sensorherstellung. -----20
- Abbildung 14 Histogramm der Anzahl von Leerungen bei verschiedenen Füllständen für die drei
Szenarien „Status-Quo“, „optimierte Sammlung“ und „optimierte Sammlung mit
Kapazitätsanpassung“. (Abbildungen: Zolitron)-----23

1 Einleitung

Die von der Firma Zolitron entwickelten Sensoren zur Füllstandsmessung erlauben es den Füllstand von Sammelcontainern wie z.B. Altglascontainer automatisch, digital zu erfassen. Der aktuelle Füllstand und die Position der Behälter können damit im betrieblichen Planungsprozess berücksichtigt werden. Möglich ist eine Anpassung der Tagestour auf Basis der Füllstandswerte, also das Auslassen leerer oder das Hinzunehmen voller Container. Auch eine digitale Routenplanung unter zusätzlicher Berücksichtigung von Durchfahrtshöhen, Rückwärtsfahrkataster, Vermeidung von Linksabbiegen und aktueller Verkehrslage kann implementiert werden. So kann tagesaktuell die optimale Route berechnet werden. Zusätzlich ist eine Optimierung der aufgestellten Container auf Grundlage der gemessenen Daten möglich um die Leerungsgebiete zu homogenisieren und die Leerungen effizienter zu gestalten.

Die USB Bochum GmbH (USB) hat diese Sensoren flächendeckend an ihren Altglas-Sammelcontainer installiert und misst so die Füllstände der Altglascontainer.

Im Rahmen der themenoffenen Projektförderung der Deutschen Bundesstiftung Umwelt wird die Technologie von den Projektpartnern Zolitron und USB weiterentwickelt. Es wird die Anbindung der IT-Systeme gezeigt und die dynamische Routenplanung entwickelt. Die entwickelten Lösungen werden dann im realen Betrieb vom USB getestet.

Durch das Wuppertal Institut findet eine Auswertung der so erhobenen Daten zum Status-Quo der Altglas-Sammlung, eine Abschätzung der CO₂ Emissionen durch die Sensoren, sowie eine Berechnung der Einsparpotenziale durch die Routenoptimierung statt.

2 Methodik

Für die Bewertung wurde der Zeitraum “Status Quo”, in dem zwar Daten erhoben werden, jedoch keinerlei Optimierung erfolgt ausgewertet. Diese wurden dann mit den Daten zum simulierten optimierten Betrieb verglichen.

2.1 Datenauswertung

Die von der USB und Zolitron erhobenen Daten liegen in drei unabhängig erfassten Datensätzen vor.

1. Glasmengen

Zur Erfassung der Glasmenge wurden folgende Variablen erhoben

- Kennzeichen des Leerungsfahrzeuges
- Zeitpunkt des Eintrags
- Glassorte (Grün, Braun oder Weißglas)
- geleerte Menge in Tonnen, erfasst bei der Leerung der Fahrzeuge

	D_KENNZEICHEN	D_DATUM	D_ZEIT	D_NR	A_NR	L_NR	D_ARTIKEL	D_MENGE
0	BO UB 224	2022-04-01	09:32:00	132006659	129000095	1030	Grünglas unaufbereitet aus dualen Systeme 0102	3.18
1	BO UB 224	2022-04-01	09:41:00	132006660	129000096	1010	Braunglas unaufbereitet aus dualen Systemen 0103	1.48
2	BO UB 224	2022-04-01	09:47:00	132006662	129000094	1040	Weißglas unaufbereitet aus dualen Systeme 0101	6.14
3	BO UB 1400	2022-04-01	10:38:00	132006883	129000094	1040	Weißglas unaufbereitet aus dualen Systeme 0101	3.96
4	BO UB 1400	2022-04-01	10:48:00	132006903	129000095	1030	Grünglas unaufbereitet aus dualen Systeme 0102	3.22

Abbildung 1 Auszug aus dem Datensatz zur Erfassung der Glasmengen.

2. Füllstandserfassung

Für die Erfassung der Füllstände wurden folgende Variablen erhoben

- Zeitpunkt der Erfassung, Messung immer zur Leerung
- Container ID
- Füllstand in %

	asset_id	name	time	fill_level
0	0008186c-5747-11ec-90ba-6accc8a6f796	400012	2022-06-10	70
1	0008186c-5747-11ec-90ba-6accc8a6f796	400012	2022-06-23	25
2	000e4b82-1de7-11eb-8c17-8e679ccb3f9	400215	2022-06-08	30
3	000e4b82-1de7-11eb-8c17-8e679ccb3f9	400215	2022-06-15	40
4	000e4b82-1de7-11eb-8c17-8e679ccb3f9	400215	2022-06-23	55
5	0076fde6-4170-11ec-b95b-6accc8a6f796	400818	2022-06-07	55
6	0076fde6-4170-11ec-b95b-6accc8a6f796	400818	2022-06-20	65
7	00ca5af0-574f-11ec-bcce-1ecacf31f3f0	400094	2022-06-03	60
8	00ca5af0-574f-11ec-bcce-1ecacf31f3f0	400094	2022-06-17	60
9	01615016-56a2-11ec-bcce-1ecacf31f3f0	Staphanstr Braun	2022-06-13	55

Abbildung 2 Auszug aus dem Datensatz zur Erfassung der Glasmengen.

3. Strecke der Sammelfahrzeuge

Für die von den Sammelfahrzeugen zurückgelegte Strecke wurden folgenden Daten erhoben:

- Kennzeichen des Leerungsfahrzeuges
- Datum
- Kilometerstand Start und Kilometerstand Ende

daraus ergibt sich unmittelbar die zurückgelegte Strecke

	Datum	Kennzeichen	Anfang	Ende	Kilometer
0	2022-04-01	224.0	139970.0	140140.0	170.0
1	2022-04-02	224.0	140140.0	140200.0	60.0
2	2022-04-03	NaN	NaN	NaN	NaN
3	2022-04-04	224.0	140200.0	140342.0	142.0
4	2022-04-05	224.0	140342.0	140449.0	107.0

Abbildung 3 Auszug aus dem Datensatz zur Erfassung der Strecken.

Die Daten wurden mit Hilfe der Programmiersprache Python eingelesen, verarbeitet, visualisiert und ausgewertet. Der komplette Source-Code mit einem erklärenden Text findet sich im Anhang. Alle nicht hier im Methodenteil erläuterten Berechnungswege können dort im Detail nachvollzogen werden.


```
[ ] # Visualisiere die Glasmenge der Container bei Leerung im zeitlichen Verlauf für die drei Glassorten
fig, ax = plt.subplots(figsize=(16,9))

ax.tick_params(labelsize=14)
colors = {1030:'forestgreen', 1010:'saddlebrown', 1040:"#006D72"}
labels = {1030:'Grünglas', 1010:'Braunglas', 1040:'Weißglas'}

grouped = df.groupby('L_NR')
for key, group in grouped:
    group.plot.scatter(ax=ax, x='D_DATUM', y='D_MENGE', label=labels[key], color=colors[key], s=15)
ax.grid(linewidth=0.5)
ax.set_xlabel("Datum", size=16, labelpad=10)
ax.set_ylabel("Menge in Tonnen", size=16, labelpad=10)
plt.title("Glasmenge der Container im zweiten Quartal", fontsize=16);
plt.show()
```

Abbildung 4 Beispiel für die Auswertung der Daten, hier Erstellen eines Diagramms zur Abbildung der Glasmengen, mit der Glasmenge auf der Y-Achse, der Zeit auf der X-Achse sowie farblicher Codierung der Datenpunkte zur Unterscheidung von Braun-, Grün- und Weißglas .

2.2 Routenoptimierung

Für die Routenoptimierung wurde die Daten der Füllstandssensoren zu Füllständen und Leerungen verwendet um die Länge der Füllzyklen aller Behälter zu bestimmen. Auf Basis dieser Daten wurde dann eine statistische Analyse der Füllgeschwindigkeiten durchgeführt und die Behälter nach Standorten geclustert. Unter Berücksichtigung von Behälterbewegungen aus den GPS Daten der Sensoren können so für jeden Standort Füllgeschwindigkeiten berechnet werden.

Im nächsten Schritt wurde eine Distanzmatrix für die Tourenplanung berechnet. Hierzu wurde open Streetmap Kartenmaterial und die open source routing engine „Valhalla“ verwendet. Rahmenbedingungen waren unter anderem „Leerung von der korrekten Straßenseite aus“, „präferieren von Rechtsabbiegen“ und „keine U-Turns“. Höhen- und Gewichtsbeschränkungen auf der Strecke und historische Verkehrsdaten wurden mit einbezogen.

Für die Tourenplanung wurden außerdem die Kapazität der Sammelfahrzeuge anhand von historischen Daten und die Arbeitszeiten berücksichtigt. Startpunkt für die jeweiligen Arbeitszeitfenster waren die USB Bochum, Endpunkt die Verwertungsanlage in Essen.

Es wurde dabei ein bestimmter Planungszeitraum, z.B. acht Wochen, betrachtet, für den dann Aufträge generiert werden, wie in dem folgenden Beispiel erklärt:

Ein Standort der sich alle 18 Tage füllt muss bei einer zyklischen, 4-wöchigen Planung, 2 Mal in 4 Wochen geleert werden. Das heißt für diesen Zeitraum werden zwei Aufträge generiert.

Alle diese Eckdaten wurden als Bedingung genutzt um eine Route zu finden, bei der die Fahrzeit minimal ist. Dazu wurden die Bedingung in ein Constrained Programming Problem überführt, das mit Google OR Tools als Constrained Solver gelöst wurde.

2.3 Berechnung der CO₂ Emissionen

Ziel dieser Studie war es, die Veränderung der CO₂-Emissionen zwischen zwei betrachteten Szenarien zu erfassen, einmal dem Status Quo der Abfallsammlung und einmal mit einer Routenoptimierung auf Basis der Füllstandsdaten. Als Vergleichsindikatoren bieten sich die spezifischen Größen „CO₂-Emissionen pro Tonne

entsorgtes Glas” oder “CO₂-Emissionen pro Zeit” an. Aufgrund der verfügbaren Daten wurde hier, für die Auswertung, die “CO₂-Emissionen pro Zeit” verwendet.

2.3.1 CO₂ Emissionen durch die Herstellung und den Betrieb des Sensors

Um die CO₂-Emissionen zu berechnen, die bei der Herstellung eines Sensors anfallen, wird zunächst der Aufbau des Sensors genauer betrachtet. In Abbildung 5 sind die verschiedenen Komponenten eines Füllstandssensors der Firma Zolitron dargestellt.

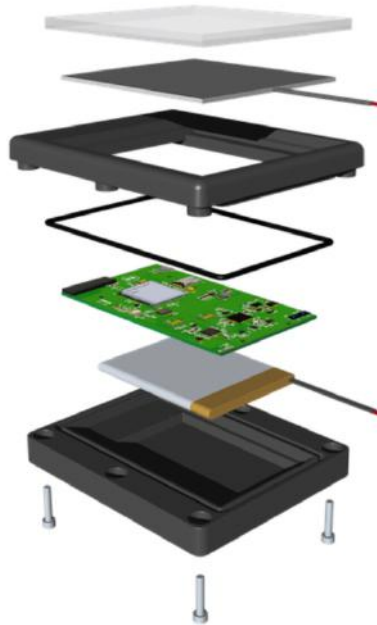


Abbildung 5: Aufbau Füllstandssensor mit Deckplatte, Solarzelle, Gehäuseoberseite, Dichtung, Platine, Akku, Gehäuseunterseite und Schrauben (Abbildung: Zolitron).

Nach Rücksprache mit der Firma Zolitron wird der Sensor in die in der untenstehenden Tabelle aufgelisteten Komponenten und Gewichtsangaben unterteilt und dementsprechend untersucht.

Tabelle 1 Liste der Komponenten des Füllstandssensors zur Ermittlung der CO₂ Emissionen.

Nr.	Komponente	Gewicht [g]	Kommentar
1	Solarzelle	5	Gewichtsangabe durch Zolitron
2	Leiterplatte (inkl. Komponenten)	5	Gewichtsangabe durch Zolitron
3	Akku	60	Gewichtsangabe durch Zolitron; Batterie-Technik nicht bekannt, hier mit Li-Ionen Akku berechnet
4	Gehäuse ABS (ohne Schrauben)	120	Gewichtsangabe durch Zolitron
5	Schrauben: sechs T6 und zwei T2, beide aus Edelstahl	30	Gewicht näherungsweise angenommen

Der Herstellungsprozess eines Füllstandssensors wurde mithilfe des Ökobilanzierungsprogramms openLCA bilanziert. Die Berechnungen wurden unter Verwendung

der Daten aus ecoinvent v3.7.1 durchgeführt und die unter dem Eintrag “transport, freight, lorry 16-32 metric ton, EURO6 | APOS, U -RER.” hinterlegten Daten für den Transport des Altglases verwendet. Die Daten zur Berechnung der CO₂ Emissionen für die Herstellung der Sensoren wurden mit folgenden Daten aus der ecoinvent Datenbank berechnet, die jeweiligen Massen sind in Tabelle 1 angegeben:

- Elektronik – printed wiring board, surface mounted, unspecified, Pb free
- Solarzelle - photovoltaic cell, single-Si wafer
- Schrauben - forging steel
- Gehäuse - acrylonitrile-butadien-styrene copolymer
- Akku - battery cell, Li-ion

Die Emissionen durch den Betrieb der Sensoren, wurden mit Hilfe weiterer Literaturdaten abgeschätzt. Es wurden folgende Quellen verwendet:

- Leistung eines Cloud-Speichers (Gröger, 2020, S. 30)
- Energieaufwand für die Übertragung der Daten an das Endgerät (Gröger, 2021, S. 101)
- CO₂-Intensität des deutschen Strommixes (Icha, 2023, S. 25)

3 Ergebnisse und Diskussion

3.1 Daten zum Status Quo

3.1.1 Glasmenge

Die Glasmenge wurde für zwei Sammelfahrzeuge erfasst, dabei wurde nach Braun-, Grün- und Weißglas unterschieden. Insgesamt wurden 391 Leerungen im zweiten Quartal 2022 (01.04.2022 - 31.06.2022) erfasst. Die Daten sind in der folgenden Grafik visualisiert.

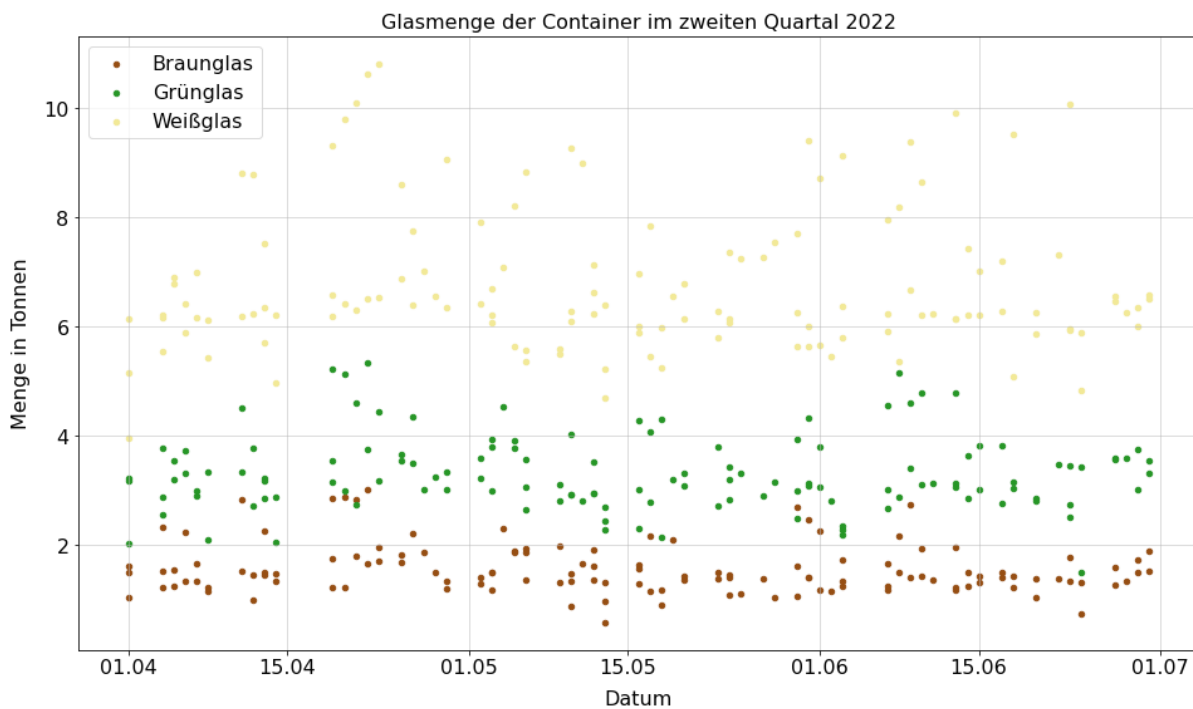


Abbildung 6 Visualisierung der erfassten Glasmenge.

Wie aus der oben abgebildeten Grafik, Abbildung 6, hervorgeht variiert die Glasmenge bei der Leerung zwischen weniger als 1 t und etwas mehr als 10 t Glas. Durch die Einfärbung der Datenpunkte entsprechend ihrer Glassorte wird deutlich, dass die erfassten Mengen für Braunglas am geringsten sind. Sie liegen zum Großteil zwischen 1 t und 2 t. Für Grünglas wird etwas mehr Glas pro Leerung erfasst meist um die 3 t. Die Weißglasmengen pro Leerung beträgt oft ca. 6 t, es gibt aber auch Leerungen mit deutlich mehr erfasstem Weißglas, bis knapp über 10 t. Noch besser zeigt sich diese Verteilung der Daten im folgenden Diagramm, Abbildung 7. Hier wurden die Daten nochmal als Histogramm visualisiert. Aus dieser Verteilung lässt sich noch nicht ableiten, ob es hier Optimierungspotential für die Leerung gibt. Eine breite Streuung der Massen kann verschiedene Ursachen haben

- verschiedene Containergrößen. Bei den Containern des vorliegenden Datensatzes wurden 3,2 m² Container gemessen. Allerdings finden sich auch geteilte Braun- und Grünglas Container unter den gemessenen Containern.
- verschiedene Anzahl von Containern. Je nach Stellplatz können ein oder zwei Container für Stellplätze vorhanden sein und Braun- und Grünglas in geteilten oder einzelnen Containern gesammelt werden.

- verschieden dicht gelagertes Füllgut in den Containern
- verschiedene Füllstände bei der Leerung. Dies spräche dann für Optimierungspotential durch eine Füllstandsmessung

Für mehr Einsicht zum Optimierungspotential gibt es zwei Möglichkeiten. Zum einen die zusätzliche Erfassung der jeweiligen Containergrößen sowie deren Anzahl. Mit Hilfe dieser Daten ließe sich die Masse höchstwahrscheinlich mit den Füllständen der Container korrelieren. Alternativ bietet sich die von Zolitron durchgeführte Füllstandsmessung an, hier werden direkt die Füllstände erfasst.

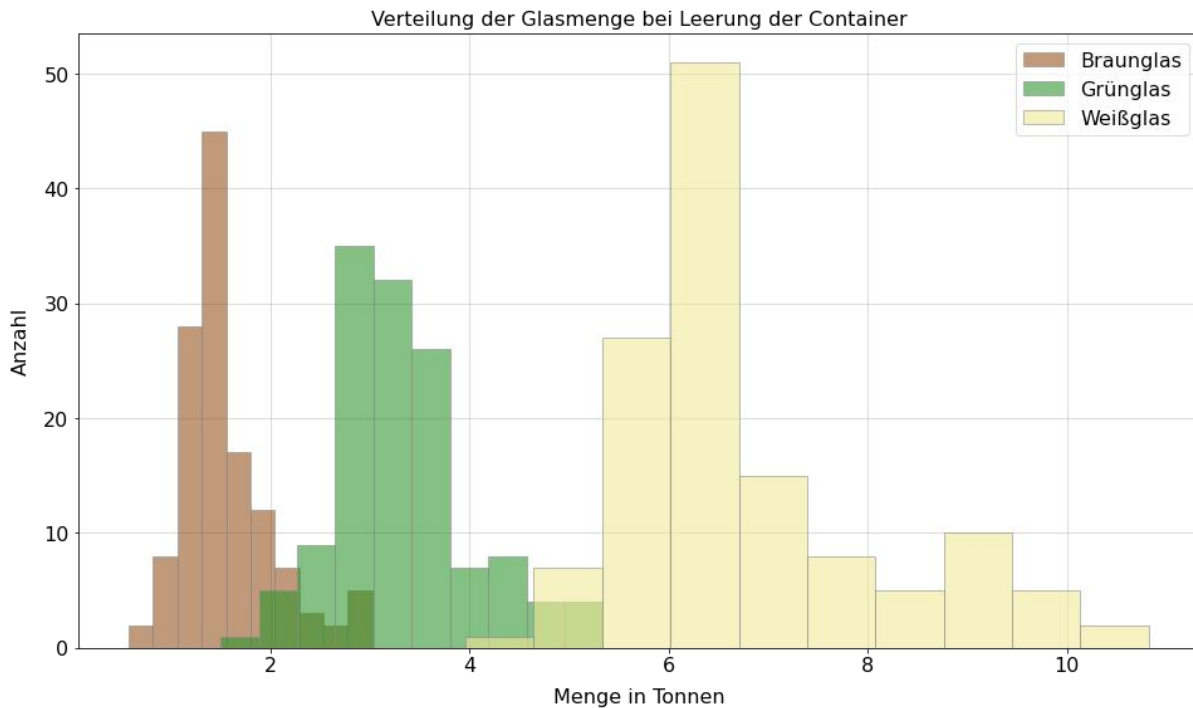


Abbildung 7 Histogramm der Glasmengen zur Veranschaulichung der verschiedenen Maxima und Breiten der jeweiligen Verteilung.

Auch die Variation der kumulierten täglichen Glasmenge lässt sich aus dem vorhandenen Datensatz ableiten und ist im untenstehenden Diagramm, Abbildung 8, visualisiert. Wie zu erwarten ist die tägliche Gesamtmenge an Glas stark abhängig von der Anzahl der durchgeführten Leerungen und liegt zum Großteil im Bereich zwischen 20 und 30 t/d. Die Anzahl der täglichen Leerungen bewegt sich meistens im Bereich zwischen 6 und 9 Leerungen.

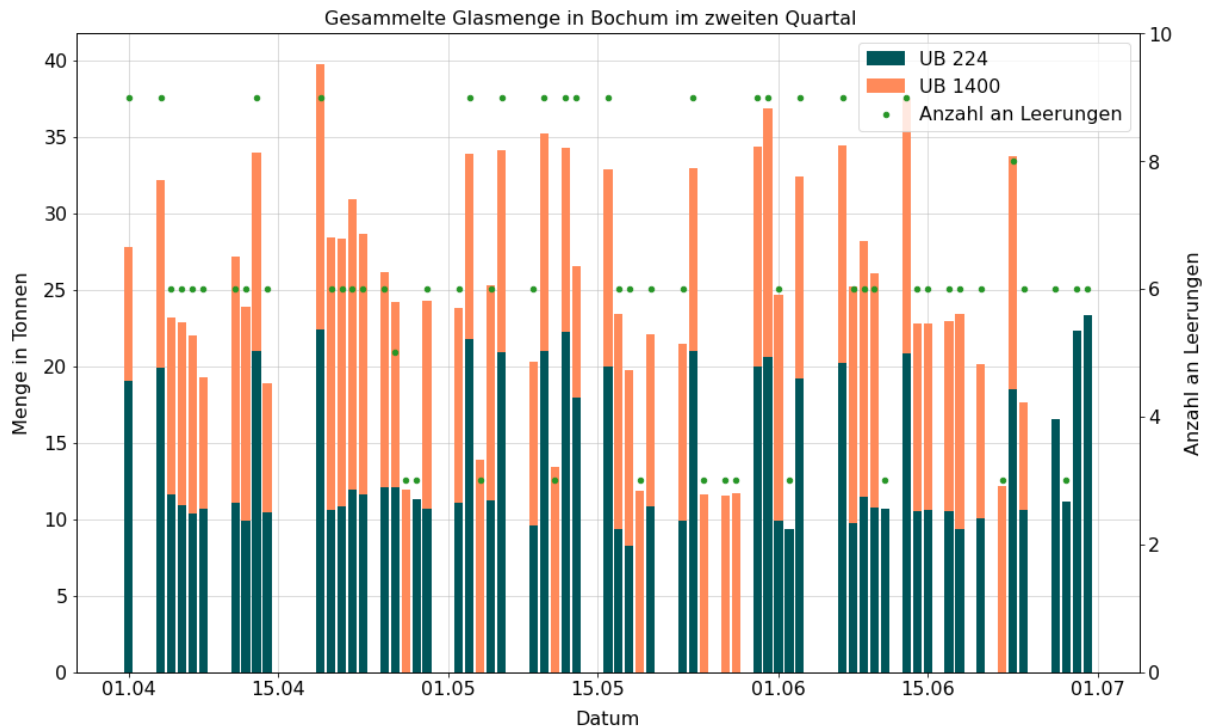


Abbildung 8 Kumulierte tägliche Glasmenge und Anzahl der Leerungen für beide Fahrzeuge.

3.1.2 Füllstandserfassung

Die Daten zur Füllstandserfassung wurden im Zeitraum Juni 2022 erhoben, es wurden 1018 Füllstandsmessungen zu 23 verschiedenen Zeitpunkten an 542 Containern durchgeführt. Die Messungen erfolgten jeweils zum Zeitpunkt der Leerung, bzw. in den 6 h davor. Zu einem gegebenen Container liegen allerdings nur maximal 5 Datenpunkte vor. Die Füllstände der Container sind in der untenstehenden Grafik, Abbildung 9, als Histogramm abgebildet. Die relative breite Verteilung der Füllstände zeigt das bestehende Optimierungspotential auf. Aus der kumulierten Häufigkeitsverteilung lässt sich ablesen dass 39 % der Leerungen an Containern mit einem Füllstand von 50 % und weniger stattfanden. In der folgenden Tabelle ist, zum besseren Verständnis, diese kumulierte Häufigkeitsverteilung noch einmal verschriftlicht. Die niedrigen Füllstände sind teilweise den Fraktionen Grün- und Braunglas zuzuordnen, die meist bei der Abholung einen niedrigeren Füllstand haben. Durch eine optimierte Routenplanung ließe sich hier die Verteilung verschmälern und hin zu höheren Füllständen verschieben. Das spart zum einen Arbeits- und Fahrzeugzeiten und kann zum anderen auch die zurückgelegte Strecke mindern.

Tab. 3-1 Tabelle zur Verteilung der Füllstände bei Leerung der Container

Füllstand bei Leerung	Anteil an Gesamtleerung
0 - 10 %	0,8 %
0 - 20 %	5,0 %
0 - 30 %	13,1 %
0 - 40 %	25,1 %
0 - 50 %	38,7 %
0 - 60 %	52,4 %
0 - 70 %	66,7 %
0 - 80 %	80,6 %
0 - 90 %	91,4 %

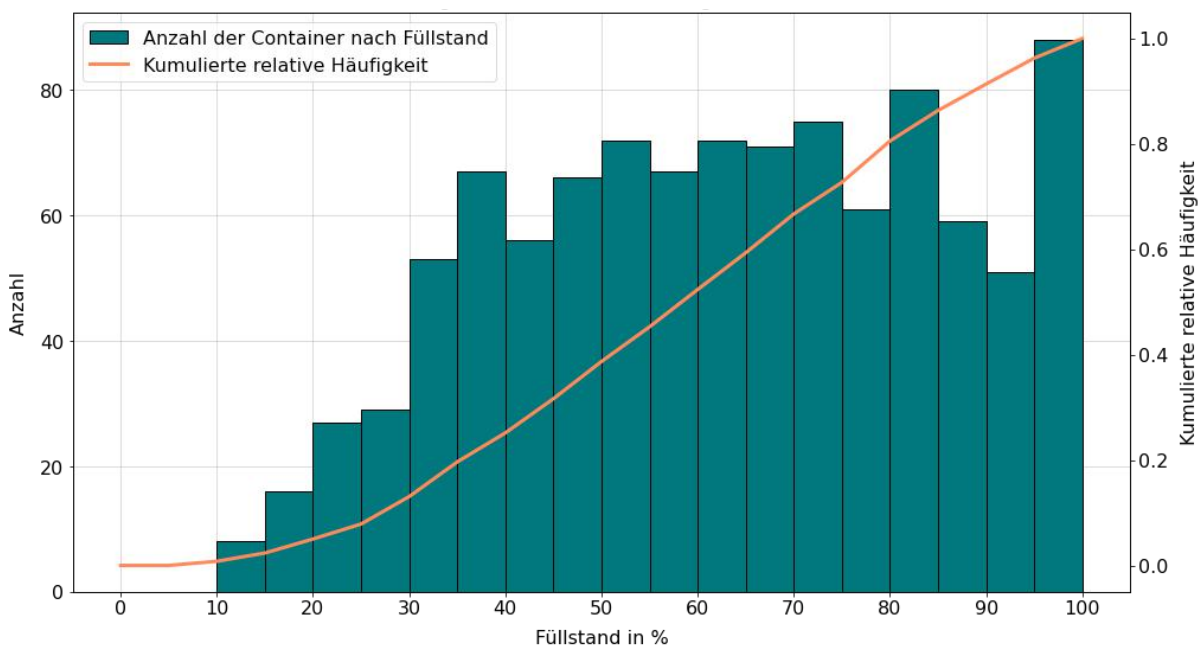


Abbildung 9 Verteilung der Füllstände bei der Leerung der Container.

Eine Aufbereitung der Daten für individuelle Container findet sich in Abbildung 10. Hier wurde der Füllstand zur Leerung für alle Container mit mehr als 3 Leerungen geplottet. Auch wenn die geringe Menge an Datenpunkten eine Aussagekräftige Interpretation erschwert, da die Streuung statistischer Natur sein kann, scheint es einige Container zu geben, die zur Leerung zu über 60 % gefüllt sind, wie z.B. Container Nummer 2, 3, 4, 11 in der Abbildung 10. Andere Container weisen eine größere Variation der Füllstände auf, wie z.B. Container 8, 14, oder 17. Es gibt allerdings kaum Container, die bei jeder Leerung weniger als 30 % Füllstand hatten. Falls sich dieses Bild auch über einen längeren Messzeitraum und mit mehr Datenpunkten bestätigen lässt, ergibt sich daraus eine Aussage zum Optimierungspotential: Für Container die meistens sehr voll oder überfüllt sind, kann in Erwägung gezogen werden, diese öfter anzufahren oder einen weiteren Container aufzustellen sofern dass die örtlichen Gegebenheiten zulassen. Für Container mit stark schwankender Füllmenge sollte die Optimierung bevorzugt mit einer situativ angepassten Tourenplanung auf Basis der aktuellen Füllstände erfolgen.

Das Optimierungspotential zeigt sich auch noch mal gut in der Darstellung in Abbildung 11. Hier wurden die Füllstände der Container gegen die Zeit aufgetragen. Die maximale Füllstandskurve zeigt an, dass es immer Container gab, die zu mehr als 95 % gefüllt waren. Diese Container hätten schon früher geleert werden können. Aufgrund der Daten kann nicht ausgeschlossen werden, dass die Container an einem Standort mit zwei Containern pro Glassorte unterschiedlich schnell befüllt wurden. Also, dass in Einzelfällen ein Weißglascontainer schon voll war, während daneben ein nicht voller stand. In aktuelleren Daten wird diese Einschränkung durch die Erhebung von Standortfüllständen behoben. Insgesamt wurden 13,1 % der Leerung an Containern mit ≤ 30 % Füllstand durchgeführt und 8,6 % der Container hatten bei der Leerung einen Füllstand von ≥ 95 %.

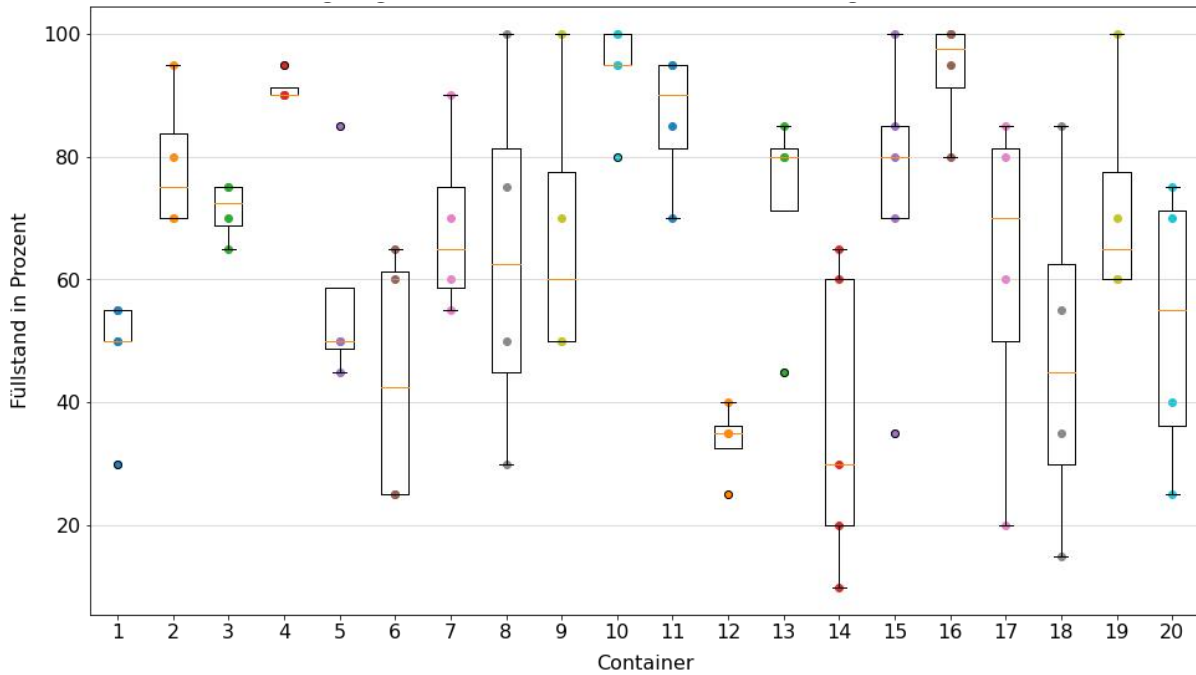


Abbildung 10 Füllstände für individuelle Container (Nummer 1 bis 20) bei Leerung für alle Container, die im Messzeitraum dreimal oder öfter geleert wurden.

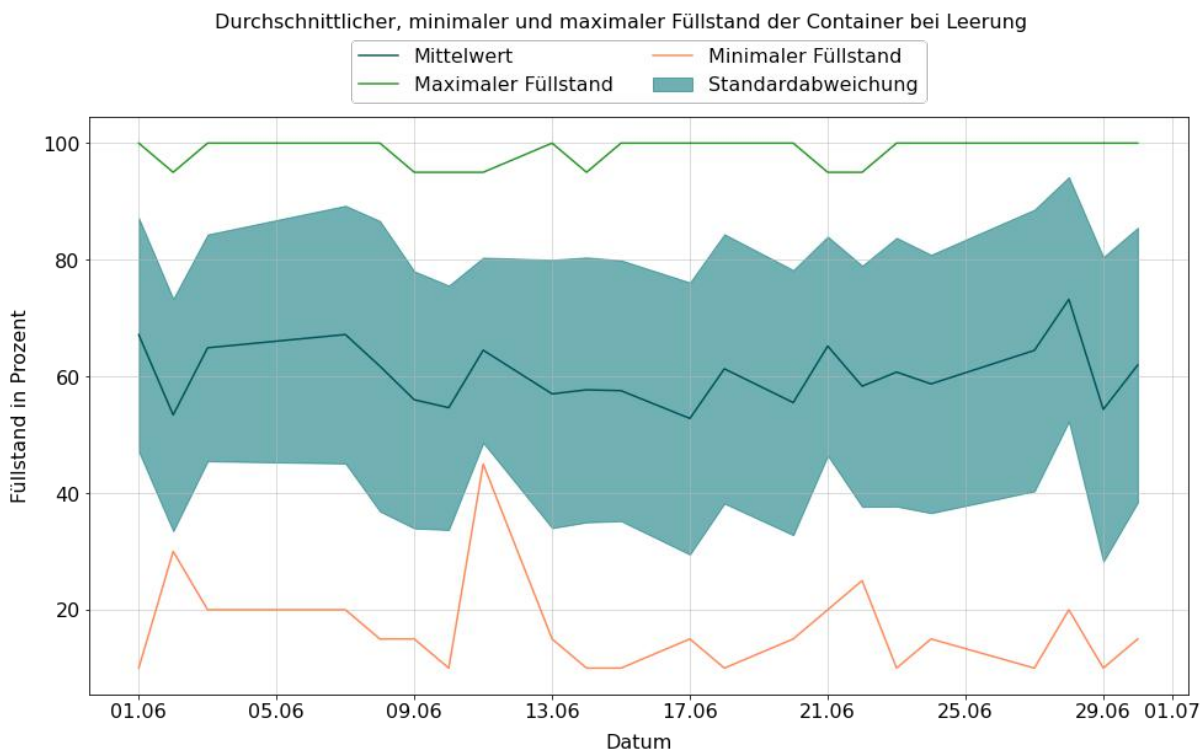


Abbildung 11 Füllstand der Container zum Zeitpunkt der Leerung, aufgetragen gegen die Zeit. Die Füllstände der Container mit höchstem und niedrigstem Füllstand an diesem Tag sind jeweils als maximaler bzw. minimaler Füllstand eingezeichnet.

3.1.3 Strecke der Sammelfahrzeuge

Die zurückgelegte Strecke der Sammelfahrzeuge wurde für 2 Sammelfahrzeuge im zweiten Quartal 2022 (01.04 - 30.06.) erfasst. Diese sind insgesamt 13060 km

gefahren. Ein Fahrzeug hat davon 6975 km zurückgelegt, das andere Fahrzeug 6085 km. Im Mittel fahren die Fahrzeuge jeweils 110 km pro Tag.

Da sowohl für die Strecke als auch für die gesammelte Glasmenge die beiden Parameter "Datum" und "Fahrzeugkennzeichen" erfasst wurden, lassen sich die erfassten Daten zusammenführen. So kann dann berechnet werden, wie viel Glas im Durchschnitt pro gefahrenem Kilometer gesammelt wurde. Falls die Routenoptimierung durch die Füllstandsmessung dazu führt, dass für dieselbe Glasmenge weniger Kilometer zurückgelegt werden müssen, ist davon auszugehen, dass sich das in einer Minderung des Spritverbrauchs und der Treibhausgasemissionen niederschlägt.

Die Auswertung zeigt, dass im Durchschnitt 0,113 t Glas pro km gesammelt wurden. Die Daten sind in Abbildung 12 visualisiert.

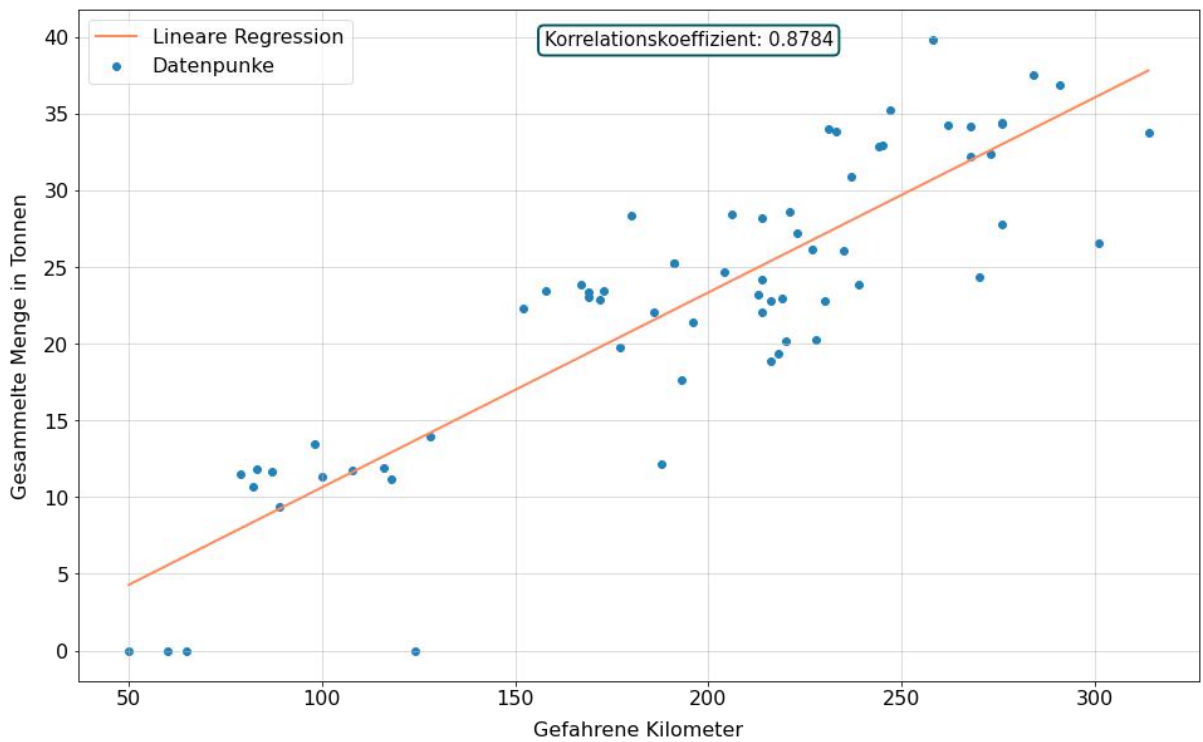
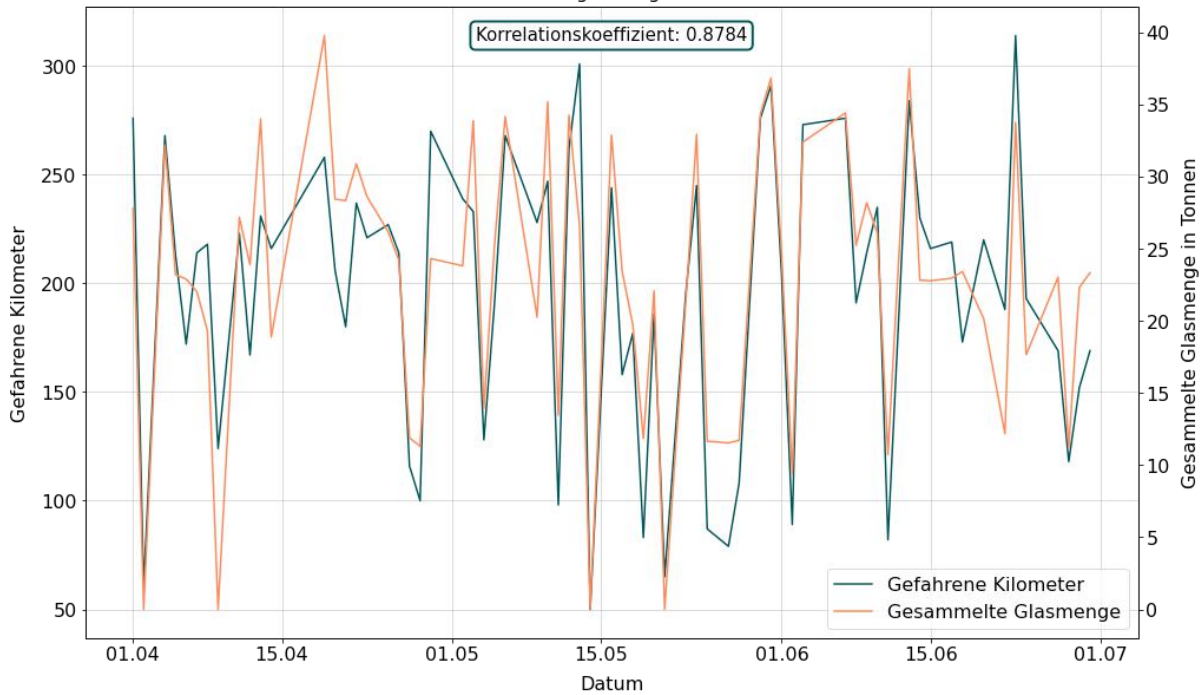


Abbildung 12 Zurückgelegte Strecke sowie gesammelte Glasmenge der beiden Sammelfahrzeuge aufgetragen gegen die Zeit.

3.2 Berechnung der CO₂ Emissionen

Für die Abschätzung der CO₂ Emissionen durch die Verwendung der Füllstandsensoren wurden folgende Aspekte berücksichtigt, die im Folgenden näher erläutert werden.

- Emissionen durch die Herstellung der Sensoren
- Emissionen durch den Betrieb der Sensoren

- Emissionen durch die Datenverarbeitung und Speicherung
- Emissionen durch den Bedarf an weiterer Hardware am Sammelfahrzeug bzw. beim Entsorger

3.2.1 Emissionen durch die Herstellung der Sensoren

Die Emissionen für die Herstellung der Sensoren wurden mit Hilfe der Software „openLCA“ und Daten aus der Datenbank „ecoinvent“ abgeschätzt. Folgende Komponenten des Sensors wurden dabei berücksichtigt

- Leiterplatte mit Komponenten
- Solarzelle
- Gehäuse
- Akku
- Schrauben

Name	Kategorie	Wirkungsergebnis	Einheit
climate change - GWP 100a		2,91351	kg CO2-Eq
> P acrylonitrile-butadiene-styrene copolymer production acrylonitrile-butadiene-styrene c	201:Manufacture of basic chemicals, fertilizers and ni...	0,45599	kg CO2-Eq
> P hard coal mine operation and hard coal preparation hard coal Cutoff, U - CN	051:Mining of hard coal / 0510:Mining of hard coal	0,09625	kg CO2-Eq
> P electricity production, hard coal, conventional electricity, high voltage Cutoff, U - ZA	351:Electric power generation, transmission and distri...	0,07679	kg CO2-Eq
> P heat production, at hard coal industrial furnace 1-10MW heat, district or industrial, othe	353:Steam and air conditioning supply / 3530:Steam ...	0,07180	kg CO2-Eq
> P electricity production, hard coal electricity, high voltage Cutoff, U - CN-NM	351:Electric power generation, transmission and distri...	0,05842	kg CO2-Eq
> P electricity production, hard coal electricity, high voltage Cutoff, U - CN-JS	351:Electric power generation, transmission and distri...	0,04396	kg CO2-Eq
> P electricity production, hard coal electricity, high voltage Cutoff, U - CN-SD	351:Electric power generation, transmission and distri...	0,04381	kg CO2-Eq
> P electricity production, natural gas, conventional power plant electricity, high voltage C	351:Electric power generation, transmission and distri...	0,04061	kg CO2-Eq
> P electricity production, hard coal electricity, high voltage Cutoff, U - US-RFC	351:Electric power generation, transmission and distri...	0,03646	kg CO2-Eq
> P diesel, burned in building machine diesel, burned in building machine Cutoff, U - GLO	431:Demolition and site preparation / 4312:Site prepa...	0,03416	kg CO2-Eq
> P electricity production, hard coal electricity, high voltage Cutoff, U - CN-SX	351:Electric power generation, transmission and distri...	0,03348	kg CO2-Eq
> P electricity production, hard coal electricity, high voltage Cutoff, U - CN-HE	351:Electric power generation, transmission and distri...	0,03274	kg CO2-Eq
> P electricity production, hard coal electricity, high voltage Cutoff, U - US-SERC	351:Electric power generation, transmission and distri...	0,03190	kg CO2-Eq
> P electricity production, hard coal electricity, high voltage Cutoff, U - JP	351:Electric power generation, transmission and distri...	0,03154	kg CO2-Eq
> P heat and power co-generation, natural gas, conventional power plant, 100MW electrical	351:Electric power generation, transmission and distri...	0,02998	kg CO2-Eq

Abbildung 13 Ausgabe von openLCA zur Aufteilung der Treibhausgasemissionen auf die verschiedenen vorgelagerten Prozesse der Sensorherstellung.

Die so berechneten Emissionen für die Produktion eines Sensors beliefen sich auf 2,92 kg CO₂eq pro Sensor. Bezieht man diese Emissionen auf alle Sensoren und auf einen Zeitraum von 1 Monat und zieht die zu erwartende mittlere Nutzungsdauer von 10 Jahren in Betracht, ergeben sich für alle Sensoren Treibhausgasemissionen von 13,16 kg CO₂eq/Monat

3.2.2 Emissionen durch den Betrieb der Sensoren

Durch den Betrieb der Sensoren entstehen keine direkten Emissionen, da der benötigte Strom über die integrierte Solarzelle zur Verfügung gestellt wird.

Emissionen durch die Datenverarbeitung und Speicherung

Speicherung der Daten

Durch den Betrieb der Sensoren fallen in einem Monat pro Sensor, 1,5 MByte an Daten an.

Die spezifische Leistung eines Cloud Speichers beträgt nach (Gröger, 2020, S. 30): 30 µW/Mbyte.

Das entspricht für 1,5 Mbyte und 1 Monat: 0,0324 Wh. Geht man von der aktuellen CO₂ Intensität des deutschen Stroms von 475 g CO₂eq/kWh aus (Icha, 2023, S. 25),

ergeben sich daraus Emissionen pro Sensor von 0,015 g CO₂eq, bzw. für alle 542 Sensoren 8,34 g CO₂eq.

Übertragung der Daten

(Gröger, 2021, S. 101) gibt den Energieaufwand zur Datenübertragung im LTE-Netz bei einer mittleren Auslastung von 1 GB/h mit 9,138 W an. Damit ergibt sich für die Übertragung der 1,5 MB vom Cloudspeicher zu den Endgeräten ein Energieaufwand von 0,0137 Wh. Nach Angaben von Zoliton findet täglich eine Übertragung der Daten vom Cloudspeicher zu den Endgeräten statt. Damit entstehen für die einmonatige Nutzungsphase und täglicher Abrufung der Daten 0,41 Wh Energieaufwand.

Mit der aktuellen CO₂ Intensität des deutschen Stroms von 475 g CO₂eq/kWh (Icha, 2023, S. 25), ergeben sich daraus Emissionen pro Sensor von 0,16 g CO₂eq für den betrachteten Zeitraum Juni 2023, bzw. rund 88 g CO₂eq für alle Sensoren.

Emissionen durch den Bedarf weiterer Hardware für die Sammelfahrzeuge

Es besteht kein Bedarf an weiterer Hardware in den Sammelfahrzeugen. Die zur Präsentation der Daten an die Fahrer der Sammelfahrzeuge verwendeten Endgeräte sind dort bereits vorhanden.

3.2.3 Emissionsausgleich durch kürzere Strecken der Sammelfahrzeuge

Die Emissionen aus anderen Bereichen sind also gering gegenüber den Emissionen aus der Herstellung der Sensoren. Demgegenüber stehen potentielle Einsparungen von Emissionen durch den LKW-Transport der Glasabfälle in Höhe von 0,162 kg CO₂eq/(t*km). Das heißt, ab einer monatlichen Einsparung von 82 t*km wirkt sich die Verwendung der Sensoren positiv auf die CO₂-Bilanz der Entsorgung aus.

3.3 Routenoptimierung

Mit Hilfe der gesammelten Sensor-Daten konnten verschiedene Routenoptionen mit optimierter Sammlung simuliert werden. Aus der Simulation ergeben sich die in Tabelle 2 dargestellten Daten zu den optimierten Routen, hier als Vergleich auf Grundlage der Daten für Juli 2023. In einem weiteren Optimierungsschritt können die Kapazitäten der Sammelplätze angepasst werden. So werden an Sammelplätzen mit hohem Altglasaufkommen mehr Container bereitgestellt während an anderen Stellen mit wenig Altglasaufkommen die Containerzahl evtl. reduziert werden kann. Simulationsdaten hierzu finden sich in der untenstehenden Tabelle 2.

Tabelle 2 Vergleich des Status Quo mit einer optimierten Route für die „optimierte Sammlung“ und für die „optimierte Sammlung mit Kapazitätsanpassung“

	Status-Quo für Juli 2023	Optimierte Sammlung	Optimierte Sammlung mit Kapazitätsanpassung
Tageskilometer [km/d]	199	156,5	149,2
Container Füllstand im Durchschnitt [%]	61,1	67,0	73,7
Leerung bei >95% Füllstand [%]	13,5	4,4	4,2
Leerungen pro Woche	638	676	700
Einsparung Treibhausgasemissionen [kg CO ₂ eq]	-13,2	2775	3254

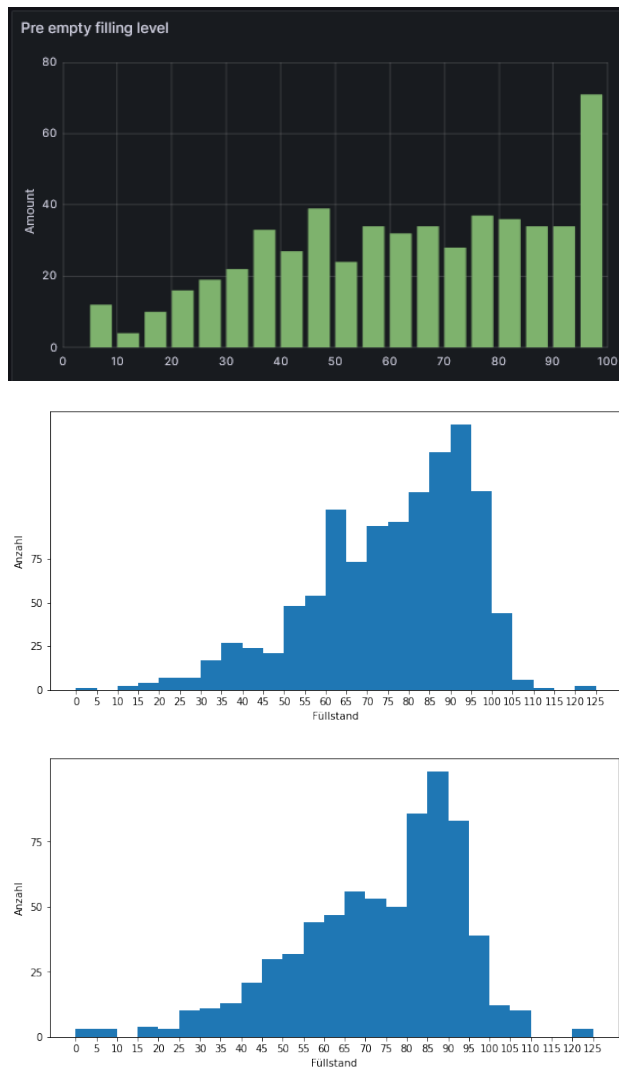


Abbildung 14 Histogramm der Anzahl von Leerungen bei verschiedenen Füllständen für die drei Szenarien „Status-Quo“, „optimierte Sammlung“ und „optimierte Sammlung mit Kapazitätsanpassung“. (Abbildungen: Zolitron)

Die Simulationsdaten zeigen klar einen Rückgang der Tageskilometer von 199 auf 156,5 bzw. 149,2 km. Das entspricht einer Einsparung von 21 % bzw. 25 %. Wie man anhand der Tabelle sieht, beruht dieser Rückgang an Kilometern nicht auf einem Rückgang der Gesamtzahl an Leerungen. Die Anzahl der Leerungen steigt sogar leicht an, von 638, auf 676 bzw. 700 Leerungen. Am Histogramm in Abbildung 14 lässt sich ablesen, dass zum einen der Anteil an Containern, die bei niedrigem Füllstand geleert werden, durch die Optimierung zurückgeht, zum anderen wird aber auch vermieden, dass Container bei Füllständen >95 % geleert werden. Diese Container mit Füllständen >95 % verursachen Beistellungen. Werden diese Container frühzeitig geleert können die Beistellungen vermindert werden und das Stadtbild wird aufgewertet. Diese aus dem Histogramm abgeleiteten Aussagen findet man auch in den Daten in Tabelle 2 wieder. Der durchschnittliche Füllstand bei der Leerung nimmt von 61,1 % auf 67,0 % bzw. 73,7 % zu. Die Zahl der Container die bei Füllständen >95 % geleert werden nimmt von 13,5 % auf 4,4 % bzw. 4,2 % ab.

Die Tageskilometerzahl lässt sich also durch eine optimierte Routenführung vermindern und durch eine Anpassung der Container-Kapazitäten lässt nochmal eine leichte Reduktion erzielen. Eine weiteres Optimierungskriterium ist die Kapazität der Sammelfahrzeuge. Diese Optimierung wird im Rahmen dieses Berichtes nicht behandelt, soll aber hier der Vollständigkeit halber kurz illustriert werden. Zurzeit befinden sich zwei Sammelfahrzeuge im Einsatz, ein großes Modell mit 19,3 t (ca. 22 Container) zulässiger Zuladung und eine kleineres Modell mit 10,7 t (ca. 11 Container) zulässiger Zuladung. Ein großer Teil der Wegstrecke der Sammelfahrzeuge entfällt auf die Fahrt zur Verwertungsanlage in Essen. Lässt sich die Anzahl dieser Fahrten verringern, verringert sich damit auch die Anzahl der Tageskilometer, dies lässt sich vor allem über die Kapazität der Sammelfahrzeuge beeinflussen.

3.3.1 CO₂ Einsparungen durch die Routenoptimierung

Die CO₂ Einsparungen aus der Routenoptimierung ergeben sich aus der Reduktion der gefahrenen Tageskilometer und dem Aufwand an Treibhausgasen für die Herstellung und Betrieb der Füllstandssensoren. Mit Emissionen von 0,162 kg CO₂eq/(t*km) für den Transport, einer mittleren Zuladung der Sammelfahrzeuge von 13,5 t sowie 13,2 kg CO₂eq Aufwand an Treibhausgasen für die Herstellung und Betrieb der Füllstandssensoren errechnen sich die in Tabelle 3 abgebildeten Einsparungen an Treibhausgasemissionen. Wie man sieht lassen sich durch die Verwendung der Füllstandssensoren 2,8 bzw. 3,3 t CO₂eq pro Monat einsparen. Zum Vergleich, ein Bewohner Deutschlands emittiert im Monat im Durchschnitt 0,93 t CO₂eq (Statista Research Department, 2022). Die relative Ersparnis an Treibhausgasemissionen durch die Verwendung der Füllstandssensoren beträgt 21 bzw. 25 %. Der Aufwand für die Herstellung und Betrieb der Sensoren ist also so gering, dass sich die der Unterschied zwischen relativer gesparter Strecke und relativ gesparten Treibhausgasen erst in den Nachkommastellen bemerkbar machen.

Es ist zu erwarten, dass diese simulierten Daten in derselben Größenordnung auch im realen Betrieb so bestätigt werden. Die Testfahrten und Datenerhebung der USB finden zum Zeitpunkt der Erstellung allerdings noch statt, sodass ein Abgleich mit diesen Daten erst in Zukunft erfolgt.

Tabelle 3 Einsparpotentiale durch die optimierte Routensammlung.

	Status-Quo für Juli 2023	Optimierte Sammlung	Optimierte Sammlung mit Kapazitätsanpassung
Gesparte Strecke [km/Monat]	0	1275	1494
Gesparte Strecke [%]	0	21	25
Gesparte THG Emissionen [kg CO ₂ eq]	-13,2	2775	3254
Gesparte THG Emissionen relativ zum Status Quo [%]	-0,1	21	25

CO₂ Preise

Geht man davon aus, dass die CO_g-Emissionen aus dem Verkehr in Zukunft bepreist werden und ähnlich teuer sind wie börslich gehandelte Emissionsrechte sind pro Tonne CO₂ zwischen 65 und 100 €/t bzw. im langfristigen Mittel ca. 85 €/t zu veranschlagen (Boerse.de, 2023). Damit ergäbe sich ein monatliches Einsparpotential von bis zu 276 Euro für den Fall der optimierten Sammlung mit Kapazitätsanpassung. Die Kosten durch den Betrieb der Sensoren werden sich also auch in einer Zukunft, mit evtl. höheren CO₂ Preisen, nicht über den CO₂-Preis, sondern über Einsparungen an Diesel, Fahrzeug- Fahrzeug- und Personalstunden amortisieren. Geht man von den gesamtgesellschaftlichen Kosten des CO₂ Preises aus, liegen diese bei 809 €/t (0 % Zeitpräferenz) (Umweltbundesamt, 2023). Damit ergibt sich eine monatliche Einsparung von 2632 € an Schäden durch den Ausstoß von Treibhausgasen.

4 Zusammenfassung und Fazit

Es wurden die Daten zum Status-Quo der Abfallsammlung ausgewertet und mit Simulationsdaten zu einer optimierten Tourenplanung verglichen. Ein beträchtlicher Teil der Container könnte mit einer optimierten Tourenplanung bei höheren Füllständen geleert werden. Ein geringer Teil, ca. 10 %, der Container ist bei der Leerung voller als 90 %. In diesen Fällen kann mithilfe der optimierten Tourenplanung eine Überfüllung, und damit Beistellungen, stark verringert werden. Die Füllstandssensoren erlauben also eine optimierte Routenplanung und ermöglichen damit die Wegstrecke der Sammelfahrzeuge zu verkürzen, in diesem Szenario bis zu 25 %. Auch überfüllte Sammelplätze können vermieden werden. Es ist zu erwarten, dass sich so auch die Arbeitsstunden des Personals und die Fahrzeugstunden reduzieren lassen.

Die CO₂ Emissionen für die Herstellung der Sensoren sowie für deren Betrieb wurden berechnet. Für die Herstellung und Betrieb der Sensoren werden im Vergleich zu den eingesparten Emissionen nur wenig Treibhausgase emittiert (0,5 %). Es lassen sich also durch die Verwendung der Füllstandssensoren Treibhausgasemissionen vermeiden, in diesem Szenario bis zu 25 %. Absolut betragen die CO₂ Einsparungen 3,3 t CO₂eq. Zum Vergleich, im gesamten Stadtgebiet Bochum werden im selben Zeitraum 215.000 t CO₂eq ausgestoßen (Stadt Bochum, o. J.).

Die Füllstandssensoren ermöglichen dem Abfallentsorger kürzere Strecken zu fahren, überfüllte Container zu vermeiden und Treibhausgasemissionen einzusparen.

5 Literaturverzeichnis

Boerse.de. (2023). *CO2 Emissionsrechte*. <https://www.boerse.de/rohstoffe/Co2-Emissionsrecht/preis/XC000AoC4KJ2>

Gröger. (2020). *Digitaler CO2-Fußabdruck Datensammlung zur Abschätzung von Herstellungsaufwand, Energieverbrauch und Nutzung digitaler Endgeräte und Dienste*. <https://www.oeko.de/fileadmin/oekodoc/Digitaler-CO2-Fussabdruck.pdf>

Gröger. (2021). *Umweltbundesamt—Green Cloud Computing Lebenszyklusbasierte Datenerhebung zu Umweltwirkungen des Cloud Computing*. https://www.umweltbundesamt.de/sites/default/files/medien/5750/publikationen/2021-06-17_texte_94-2021_green-cloud-computing.pdf

Icha. (2023). *Umweltbundesamt—Entwicklung der spezifischen Treibhausgas-Emissionen des deutschen Strommix in den Jahren 1990—2022*. https://www.umweltbundesamt.de/sites/default/files/medien/1410/publikationen/2023_05_23_climate_change_20-2023_strommix_bf.pdf

Stadt Bochum. (o. J.). *Energie- und Klimaschutzkonzept Bochum 2030*. Abgerufen 31. August 2023, von [https://www.bochum.de/C125830C0042AB74/vwContentByKey/W2BL39LR186BOCMDE/\\$File/20151109_Abschlussbericht_EKK_BO_final-red.pdf](https://www.bochum.de/C125830C0042AB74/vwContentByKey/W2BL39LR186BOCMDE/$File/20151109_Abschlussbericht_EKK_BO_final-red.pdf)

Statista Research Department. (2022). *Durchschnittliche jährliche Treibhausgasbilanz pro Person in Deutschland*. <https://de.statista.com/statistik/daten/studie/1275275/umfrage/treibhausgasbilanz-pro-person/>

Umweltbundesamt. (2023). *Gesellschaftliche Kosten von Umweltbelastungen*. <https://www.umweltbundesamt.de/daten/umwelt-wirtschaft/gesellschaftliche-kosten-von-umweltbelastungen#gesamtwirtschaftliche-bedeutung-der-umweltkosten>

6 Anhang

6.1 Ausführliche Berechnung der Emissionen für den Betrieb der Sensoren

Berechnung der Emissionen für den Betrieb der Sensoren		
Übertragung		
Leistung Übertragung für 1GB/h LTE	W bzw. J/s	9,13800
Übertragungsgeschwindigkeit	GB/h	1,00000
Übertragungsgeschwindigkeit	GB/s	0,00028
Übertragungsgeschwindigkeit	MB/s	0,27778
Übertragene Datenmenge	MB	1,50000
„Übertragungszeit“ für 1,5Mb	s	5,40000
Energieaufwand für Übertragung von 1,5MB	J	49,34520
Umrechnungsfaktor J=>Wh		0,00028
Energieaufwand für Übertragung von 1,5MB	Wh	0,01371
Aufrufe pro Monat		30,00000
Monatlicher Energieaufwand	Wh/Monat	0,41121
CO ₂ Intensität deutscher Strommix 2021	g CO ₂ eq/kWh	475,00000
Emissionen für den Betrieb eines Sensors	g CO ₂ eq/Monat	0,19532
Anzahl der Sensoren		542,00000
Emissionen für den Betrieb aller Sensoren	g CO ₂ eq/Monat	105,86610

Berechnung der Emissionen für den Betrieb der Sensoren**Datenspeicherung**

spezifische Leistung des Cloudspeichers	μW/MB	30,00000
---	-------	----------

Speicherzeit	Monat	1,00000
--------------	-------	---------

Speicherzeit	h	720,00000
--------------	---	-----------

Energieverbrauch Speicherung von 1,5MB	Wh/Monat	0,03240
--	----------	---------

Emissionen für die Speicherung der Daten ein Sensor	g CO ₂ eq/Monat	0,01539
---	----------------------------	---------

Emissionen für die Speicherung der Daten alle Sensoren	g CO ₂ eq/Monat	8,34138
--	----------------------------	---------

Emissionen Gesamt für alle Sensoren pro Monat	g CO₂eq/Monat	114,20748
--	---------------------------------	------------------

6.2 Jupiter Notebook und Source Code für die Datenauswertung

Datenexploration

January 5, 2023

1 Pakete installieren und Parameter setzen

```
[1]: # Pakete importieren und Verbindung zu Google Drive herstellen
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import datetime
from matplotlib.dates import DateFormatter

from google.colab import drive
drive.mount("/content/drive")
```

Mounted at /content/drive

```
[2]: # Pfade für die Verbindung zu Google Drive
data_path="/content/drive/Shared drives/Datenexploration Zolitron/Daten/"
output_path="/content/drive/Shared drives/Datenexploration Zolitron/Output/"
# Parameter für die folgenden Visualisierungen
Datumsformat=DateFormatter("%d.%m")
ticksize=16
labelsize=16
titlesize=16
legendsize=16
```

2 Glasmengen

```
[3]: # Daten einlesen und beschreiben sowie zusätzlich nach Fahrzeugen trennen
df=pd.read_excel(data_path+"zweites-quartal-glasmengen-bochum.xlsx")
dfa=df.loc[df["D_KENNZEICHEN"]=="BO UB 224"]
dfb=df.loc[df["D_KENNZEICHEN"]=="BO UB 1400"]
df.head()
```

```
[3]:  D_KENNZEICHEN  D_DATUM  D_ZEIT      D_NR      A_NR  L_NR  \
0      BO UB 224  2022-04-01  09:32:00  132006659  129000095  1030
1      BO UB 224  2022-04-01  09:41:00  132006660  129000096  1010
2      BO UB 224  2022-04-01  09:47:00  132006662  129000094  1040
3      BO UB 1400 2022-04-01  10:38:00  132006883  129000094  1040
4      BO UB 1400 2022-04-01  10:48:00  132006903  129000095  1030
```

```
      D_ARTIKEL  D_MENGE
0  Grünglas unaufbereitet aus dualen Systeme 0102      3.18
1  Braunglas unaufbereitet aus dualen Systemen 0103      1.48
2  Weißglas unaufbereitet aus dualen Systeme 0101      6.14
3  Weißglas unaufbereitet aus dualen Systeme 0101      3.96
4  Grünglas unaufbereitet aus dualen Systeme 0102      3.22
```

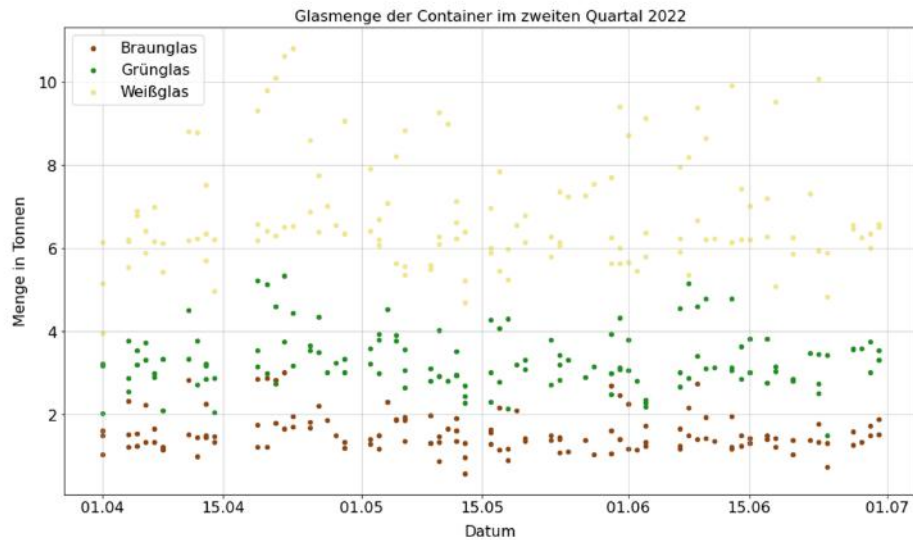
- 391 Einträge
- 2 Fahrzeuge
- 3 Glassorten (jeweils in Spalte A_NR, L_NR und D_ARTIKEL codiert)
- D_NR gibt immer einen anderen Wert, ist also als ID zu verstehen
- D_Menge wahrscheinlich in Tonnen

```
[4]: # Visualisiere die Glasmenge der Container bei Leerung im zeitlichen Verlauf
↳ für die drei Glassorten
fig, ax = plt.subplots(figsize=(16,9))

ax.tick_params(labelsize=ticksiz)
colors = {1030:'forestgreen', 1010:'saddlebrown', 1040:"khaki"}
labels = {1030:'Grünglas', 1010:'Braunglas', 1040:"Weißglas"}

grouped = df.groupby('L_NR')
for key, group in grouped:
    group.plot.scatter(ax=ax, x='D_DATUM', y='D_MENGE', label=labels[key],
↳ color=colors[key], s=20)
ax.grid(linewidth=0.5)
ax.set_xlabel("Datum", size=labelsize, labelpad=10)
ax.set_ylabel("Menge in Tonnen", size=labelsize, labelpad=10)
ax.xaxis.set_major_formatter(Datumsformat)
ax.legend(prop={'size': legendsize})
```

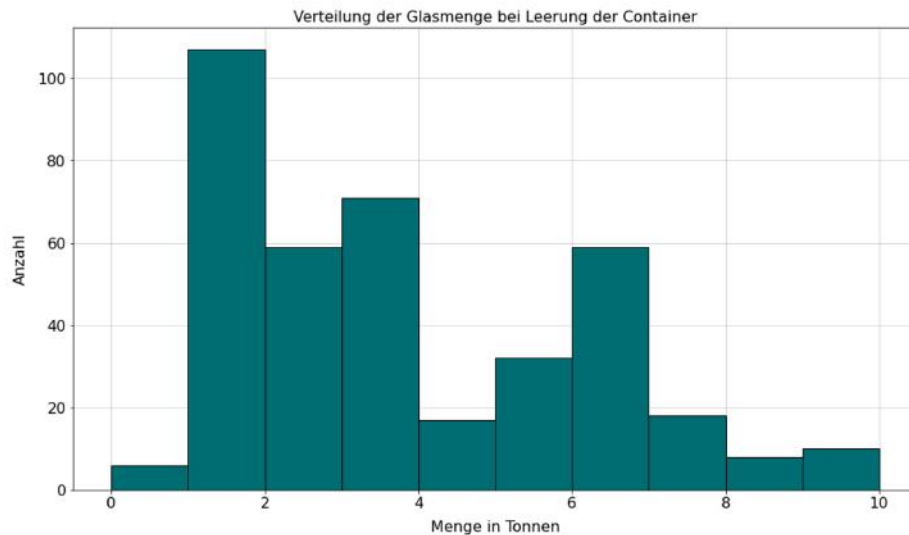
```
plt.title("Glasmenge der Container im zweiten Quartal 2022",
         font_size=title_size);
plt.show()
```



```
[5]: # Visualisiere die Verteilung der Glasmenge bei Leerung als Histogramm
fig, ax = plt.subplots(figsize=(16,9))

df.hist(ax=ax, column='D_MENGE', bins=range(11), color="#006D72", ec="black")
ax.grid(linewidth=0.5)
ax.set_ylabel("Anzahl", size=label_size, labelpad=10)
ax.set_xlabel("Menge in Tonnen", size=label_size, labelpad=10)
ax.tick_params(label_size=tick_size)
ax.set_axisbelow(True)

plt.title("Verteilung der Glasmenge bei Leerung der Container",
         font_size=title_size);
plt.show()
```

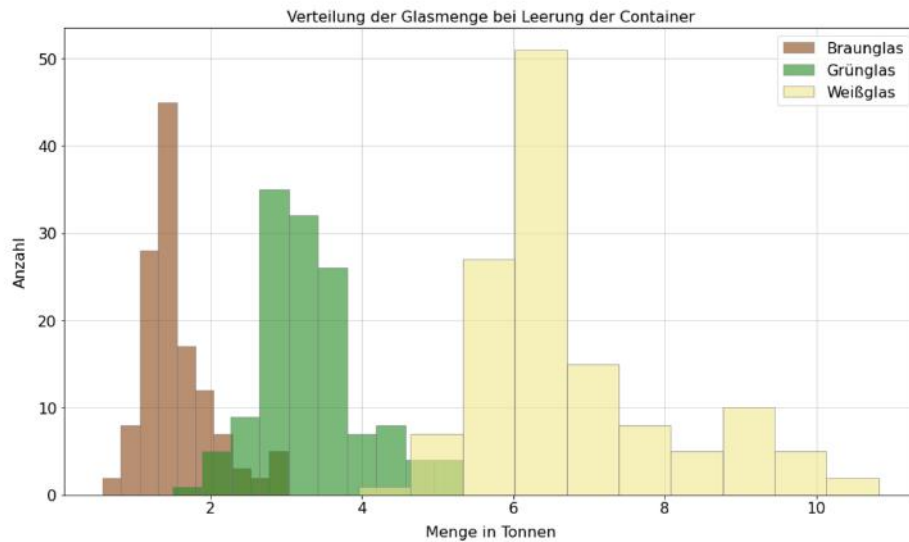
```
[6]: # Visualisiere die Verteilung der Glasmenge bei Leerung als Histogramm für die
      ↪ drei Glassorten
fig, ax = plt.subplots(figsize=(16,9))

colors = {1030:'forestgreen', 1010:'saddlebrown', 1040:"khaki"}
labels = {1030:'Grün Glas', 1010:'Braun Glas', 1040:"Weiß Glas"}

grouped = df.groupby('L_NR')
for key, group in grouped:
    group.hist(ax=ax, column='D_MENGE', label=labels[key], color=colors[key],
              ↪ ec="grey", alpha=0.6)

ax.tick_params(labelsize=ticksiz)
ax.grid(linewidth=0.5)
ax.set_ylabel("Anzahl", size=labelsize, labelpad=10)
ax.set_xlabel("Menge in Tonnen", size=labelsize, labelpad=10)
ax.set_axisbelow(True)

plt.title("Verteilung der Glasmenge bei Leerung der Container",
        ↪ fontsize=titlesiz)
ax.legend(prop={'size': legendsiz})
plt.show()
```



```
[7]: # Erstelle Daten, die den zeitlichen Verlauf der Glasmengen darstellen -> Plots
      ↳ siehe Archiv
amount_over_time=df.groupby("D_DATUM").agg({"D_MENGE":["mean", "std", "sum",
↳"count"]})
kfz_a=dfa.groupby("D_DATUM").agg({"D_MENGE":["mean", "std", "sum", "count"]})
kfz_b=dfb.groupby("D_DATUM").agg({"D_MENGE":["mean", "std", "sum", "count"]})

t=amount_over_time.index
mean=amount_over_time[amount_over_time.columns[0]]
std=amount_over_time[amount_over_time.columns[1]]
summe=amount_over_time[amount_over_time.columns[2]]
leerungen=amount_over_time[amount_over_time.columns[3]]

t_a=kfz_a.index
mean_a=kfz_a[kfz_a.columns[0]]
std_a=kfz_a[kfz_a.columns[1]]
summe_a=kfz_a[kfz_a.columns[2]]

t_b=kfz_b.index
mean_b=kfz_b[kfz_b.columns[0]]
std_b=kfz_b[kfz_b.columns[1]]
summe_b=kfz_b[kfz_b.columns[2]]

[8]: # Plote die durchschnittliche Glasmenge eines Containers bei Leerung inklusive
      ↳ der emp. Standardabweichung über die Zeit
```

```

fig, ax = plt.subplots(figsize=(16,9))

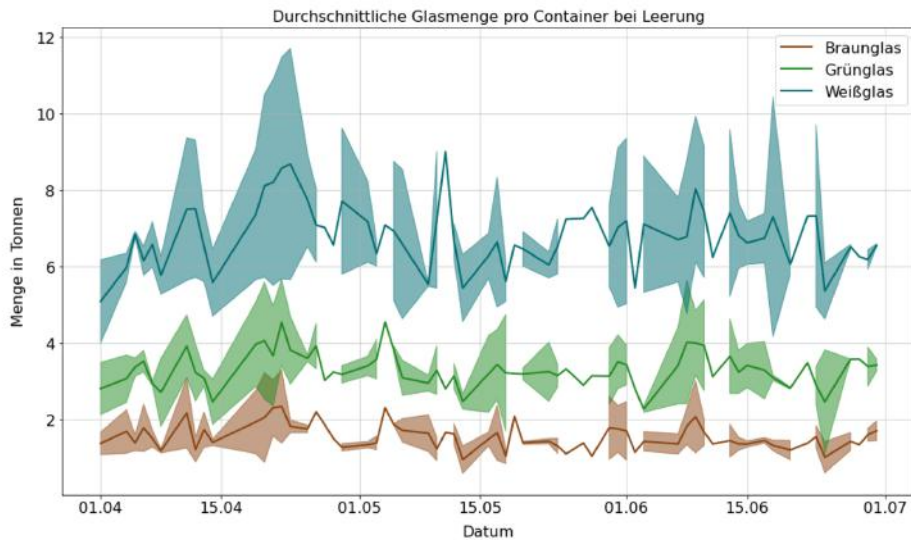
colors = {1030:'forestgreen', 1010:'saddlebrown', 1040:"#006D72"}
labels = {1030:'Grünglas', 1010:'Braunglas', 1040:'Weißglas'}

grouped = df.groupby('L_NR')
for key, group in grouped:
    time_series=group.groupby("D_DATUM").agg({"D_MENGE":["mean", "std", "sum",
    ↪"count"]})
    time_series.columns=["mean", "std", "sum", "count"]
    x=time_series.index
    M=time_series["mean"]
    std=time_series["std"]

    ax.plot(x, M, label=labels[key], color=colors[key], lw=2)
    ax.fill_between(x, M-std, M+std, color=colors[key], alpha=0.5)
    #S=time_series["sum"]
    #Leerungen=time_series["count"]

ax.grid(linewidth=0.5)
ax.tick_params(labelsize=ticksiz)
ax.set_xlabel("Datum", size=labelsize, labelpad=10)
ax.set_ylabel("Menge in Tonnen", size=labelsize, labelpad=10)
ax.xaxis.set_major_formatter(Datumsformat)
ax.legend(prop={'size': legendsize})
plt.title("Durchschnittliche Glasmenge pro Container bei Leerung",
↪fontsize=titlesize)
plt.show()

```



```
[9]: # Visualisiere den zeitlichen Zusammenhang von Anzahl der Leerungen und
↳ gesammelter Glasmenge

# Die Daten für das Fahrzeug BO UB 1400 sind nicht vollständig, daher müssen
↳ die beiden Reihen erstmal zusammengebracht werden.
A=pd.DataFrame({"time":t_a,"sum_a":summe_a})
B=pd.DataFrame({"time":t_b,"sum_b":summe_b})
A=A.join(B, how="outer", lsuffix='a', rsuffix='b')
A["sum_a"]=A["sum_a"].fillna(0)
A["sum_b"]=A["sum_b"].fillna(0)

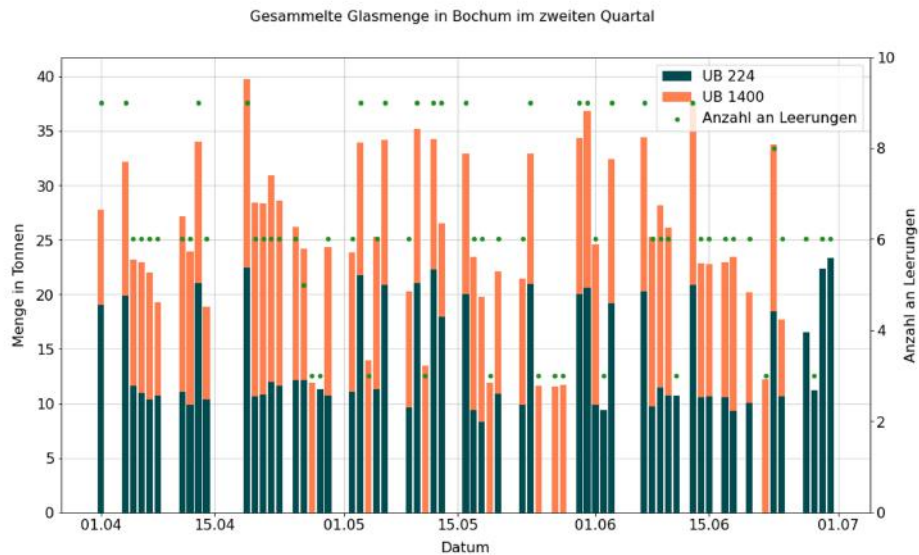
fig,ax=plt.subplots(figsize=(16,9))

y1=A["sum_a"]
y2=A["sum_b"]

ax.bar(list(A.index), y1, color="#004C50", align='center', label="UB 224")
ax.bar(list(A.index), y2, bottom=y1, color="coral", align='center', label="UB
↳1400")
ax.tick_params(labelsize=ticksiz)
ax.set_xlabel("Datum", size=labelsize, labelpad=10)
ax.set_ylabel("Menge in Tonnen", size=labelsize, labelpad=10)
ax.grid(linewidth=0.5)

# Zeichne zusätzlich die Anzahl der geleerten Container ein
ax2=ax.twinx()
ax2.tick_params(labelsize=ticksiz)
ax2.set_ylim(0,10)
ax2.set_ylabel("Anzahl an Leerungen", size=labelsize, labelpad=10)
ax2.scatter(amount_over_time.index, leerungen, c="forestgreen", s=20,
↳label="Anzahl an Leerungen")
h1, l1 = ax.get_legend_handles_labels()
h2, l2 = ax2.get_legend_handles_labels()
ax.legend(h1+h2, l1+l2, loc=1, prop={'size': legendsize})
ax.xaxis.set_major_formatter(Datumsformat)
ax.set_axisbelow(True)

plt.subplots_adjust(top=0.95)
fig.suptitle("Gesammelte Glasmenge in Bochum im zweiten Quartal",
↳fontsize=titlesize)
plt.show()
```



```
[10]: r=np.corrcoef(leerungen, summe)[0,1]
print("Die tägliche Gesamtmenge und die Anzahl der Leerungen an einem Tag_
↪weisen einen Korrelationskoeffizienten von r=%.4f auf. "%r)
```

Die tägliche Gesamtmenge und die Anzahl der Leerungen an einem Tag weisen einen Korrelationskoeffizienten von $r=0.9303$ auf.

3 Liste der Leerungen

```
[11]: l=pd.read_csv(data_path+"list-of-empties-data-2022.csv")
l["time"]=pd.to_datetime(l["time"]).dt.date # Datum zu Tag konvertieren
l.head(10)
```

```
[11]:
```

	asset_id	name	time	\
0	0008186c-5747-11ec-90ba-6accc8a6f796	400012	2022-06-10	
1	0008186c-5747-11ec-90ba-6accc8a6f796	400012	2022-06-23	
2	000e4b82-1de7-11eb-8c17-8e679ccbd3f9	400215	2022-06-08	
3	000e4b82-1de7-11eb-8c17-8e679ccbd3f9	400215	2022-06-15	
4	000e4b82-1de7-11eb-8c17-8e679ccbd3f9	400215	2022-06-23	
5	0076fde6-4170-11ec-b95b-6accc8a6f796	400818	2022-06-07	
6	0076fde6-4170-11ec-b95b-6accc8a6f796	400818	2022-06-20	
7	00ca5af0-574f-11ec-bcce-1ecacf31f3f0	400094	2022-06-03	
8	00ca5af0-574f-11ec-bcce-1ecacf31f3f0	400094	2022-06-17	

```
9 01615016-56a2-11ec-bcce-1ecacf31f3f0 Staphanstr Braun 2022-06-13
```

```
    fill_level
0         70
1         25
2         30
3         40
4         55
5         55
6         65
7         60
8         60
9         55
```

```
[12]: # Zeitliche Abdeckung der Daten ist der Juni 2022
print(min(l["time"]," - ", max(l["time"])))
```

```
2022-06-01 - 2022-06-30
```

```
[13]: print(l.shape)
l.describe(include="all")
```

```
(1018, 4)
```

```
[13]:
```

	asset_id	name	time	fill_level
count	1018	1018	1018	1018.000000
unique	542	540	23	NaN
top	d6c7bd3e-2693-11eb-97e8-b27c712d9e04	400554	2022-06-13	NaN
freq	5	5	77	NaN
mean	NaN	NaN	NaN	60.245580
std	NaN	NaN	NaN	22.754917
min	NaN	NaN	NaN	10.000000
25%	NaN	NaN	NaN	40.000000
50%	NaN	NaN	NaN	60.000000
75%	NaN	NaN	NaN	80.000000
max	NaN	NaN	NaN	100.000000

- Es gibt sehr viele (>500) verschiedene Container bzw. "Assets".
- Das Problem ist, dass für ein und denselben Namen maximal 5 Einträge vorhanden sind, d.h. eine zeitliche Analyse für einen bestimmten Container ist nur bedingt aussagekräftig.

3.1 Wie ist die Streuung der Füllmenge der Container?

```
[14]: # Visualisiere die Verteilung der Füllstände der Container als Histogramm
fig,ax=plt.subplots(figsize=(16,9))
```

```

l.hist(ax=ax, column="fill_level", color="#006D72", ec="black",
      bins=range(0,105,5), label="Anzahl der Container nach Füllstand")
ax.grid(linewidth=0.5)
ax.set_ylabel("Anzahl", size=labelsize, labelpad=10)
ax.set_xlabel("Füllstand in %", size=labelsize, labelpad=10)
ax.set_xticks(range(0,110,10))
ax.tick_params(labelsize=ticksiz)
ax.set_axisbelow(True)

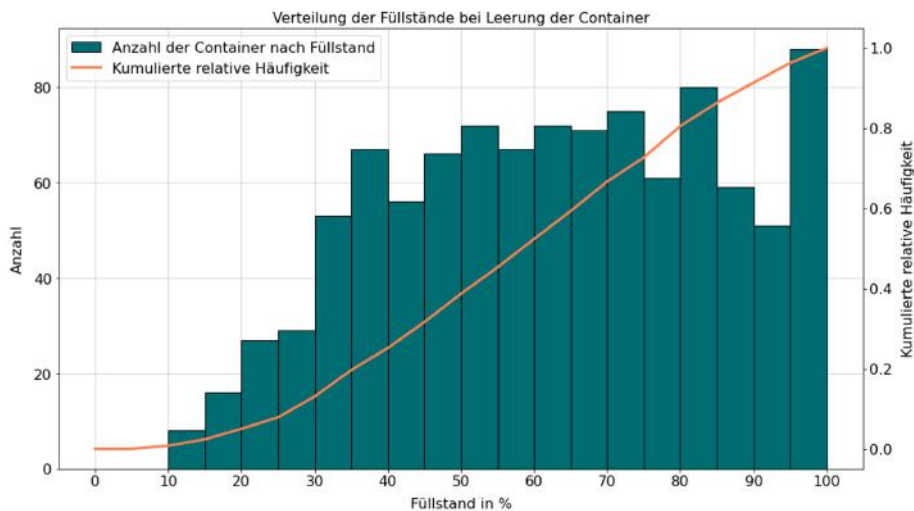
# Linienplot mit kumulierter relativer Häufigkeit
ax2=ax.twinx()
relative_freq=[]
x2=range(0,101,5)
for i in x2:
    relative_freq.append(len(l[l["fill_level"]<=i])/len(l))
ax2.plot(x2,relative_freq, c="coral", lw=3, label="Kumulierte relative
      Häufigkeit")
ax2.tick_params(labelsize=ticksiz)
ax2.set_ylabel("Kumulierte relative Häufigkeit", size=labelsize, labelpad=10)

ax.set_title("Verteilung der Füllstände bei Leerung der Container",
            fontsize=titlesize)

h1, l1 = ax.get_legend_handles_labels()
h2, l2 = ax2.get_legend_handles_labels()
ax.legend(h1+h2, l1+l2, loc=2, prop={'size': legendsize})

plt.show()

```



```
[15]: for i in range(10,100,10):
        print("0 - %d% | %i,%.1f%%"%(100*len(l[l["fill_level"]<=i])/len(l)))

print("\n", "\n", "0% - 30% | ", "%.1f%%"%(100*len(l[l["fill_level"]<=30])/
↳len(l)))
print("95% - 100% | ", " %.1f%%"%(100*len(l[l["fill_level"]>=95])/len(l)))
```

```
0 - 10% | 0.8%
0 - 20% | 5.0%
0 - 30% | 13.1%
0 - 40% | 25.1%
0 - 50% | 38.7%
0 - 60% | 52.4%
0 - 70% | 66.7%
0 - 80% | 80.6%
0 - 90% | 91.4%
```

```
0% - 30% | 13.1%
95% - 100% | 8.6%
```

```
[16]: # Untersuche die Streuung der Füllmenge für einen individuellen Container
```

```
from matplotlib.ticker import MaxNLocator
def get_N_containers_with_more_than_N_entries(N, min_data=1):
    result=[]
    Z=list(set(l["name"]))
    i=0
    k=0
    while k < N and i < len(Z):
        if len(list(l["fill_level"][l["name"]==Z[i]]))>min_data:
            result.append(Z[i])
            k=k+1
            i=i+1
    return result

def plotte_füllstand_N_container(N, min_data=1, plot_type="scatter"):
    #container=list(set(l["name"]))[0:N]
    container=get_N_containers_with_more_than_N_entries(N,min_data)
    X=[]
    Y=[]
    i=0
    if plot_type!="scatter":
        i=1
```



```

for cont in container:
    y=list(1["fill_level"][1["name"]==cont])
    x=[i]*len(y)
    if len(y) > min_data:
        X.append(x)
        Y.append(y)
        i=i+1

fig,ax=plt.subplots(1,figsize=(16,9))
if plot_type=="scatter":
    for i in range(len(X)):
        ax.scatter(X[i], Y[i])
    ax.xaxis.set_major_locator(MaxNLocator(integer=True))
if plot_type=="boxplot":
    ax.boxplot(Y)

if plot_type=="scatter und boxplot":
    for i in range(len(X)):
        ax.scatter(X[i], Y[i])
    ax.xaxis.set_major_locator(MaxNLocator(integer=True))
    ax.boxplot(Y)

ax.set_ylabel("Füllstand in Prozent", size=labelsize, labelpad=10)
ax.set_xlabel("Container", size=labelsize, labelpad=10)
ax.yaxis.grid(linewidth=0.5)
ax.tick_params(labelsize=ticksiz)

plt.title("Füllstände der Container bei Leerung\n Angezeigt werden %d
↳Container, die mindestens %d mal geleert wurden"%(len(Y),min_data),
↳fontsize=titlesize)
plt.show()

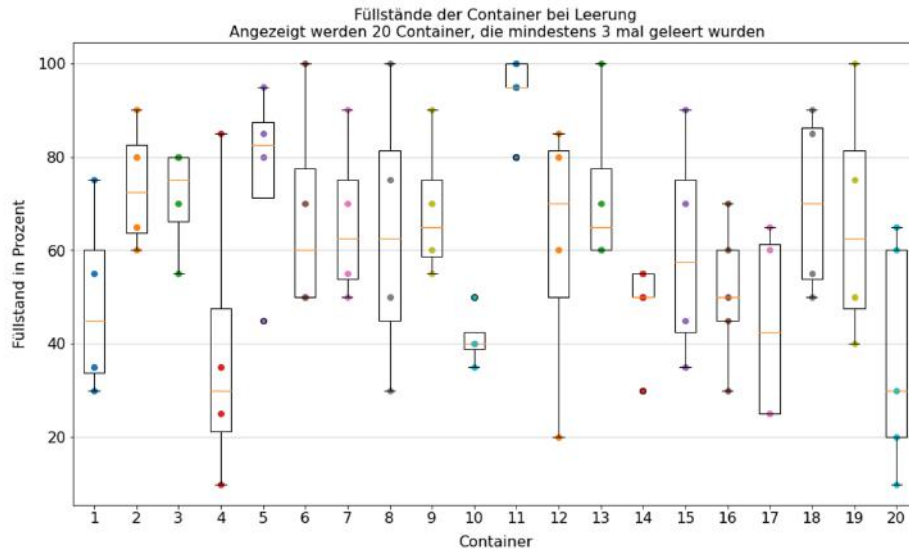
```

```
[17]: plotte_füllstand_N_container(20, min_data=3, plot_type="scatter und boxplot")
```

```

/usr/local/lib/python3.8/dist-packages/matplotlib/cbook/__init__.py:1376:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray.
X = np.atleast_1d(X.T if isinstance(X, np.ndarray) else np.asarray(X))

```



3.2 Durchschnittlicher Füllstand

```
[18]: print("Der durchschnittliche Füllstand eines Containers war im zweiten Quartal_
↳%.2f Prozent"%np.mean(l["fill_level"]))
print("Die empirische Standardabweichung beträgt dabei %.2f Prozentpunkte"%np.
↳std(l["fill_level"]))
```

Der durchschnittliche Füllstand eines Containers war im zweiten Quartal 60.25 Prozent

Die empirische Standardabweichung beträgt dabei 22.74 Prozentpunkte

```
[19]: # Entwicklung über die Zeit
fs=l.groupby("time").agg({"fill_level":["mean", "std", "sum", "count", "max", "
↳min"]})

t=fs.index
mean=fs[fs.columns[0]]
std=fs[fs.columns[1]]
#summe=fs[fs.columns[2]]
leerungen=fs[fs.columns[3]]
max_fill=fs[fs.columns[4]]
min_fill=fs[fs.columns[5]]

fig,ax=plt.subplots(figsize=(16,9))
```

```

ax.plot(t,mean, c="#004C50", label="Mittelwert")
ax.fill_between(t,mean-std,mean+std, color="#006D72", alpha=0.6,
→label="Standardabweichung")
#ax.plot(t, max_fill, c="forestgreen", label="Maximaler Füllstand")
#ax.plot(t, min_fill, c="coral", label="Minimaler Füllstand")

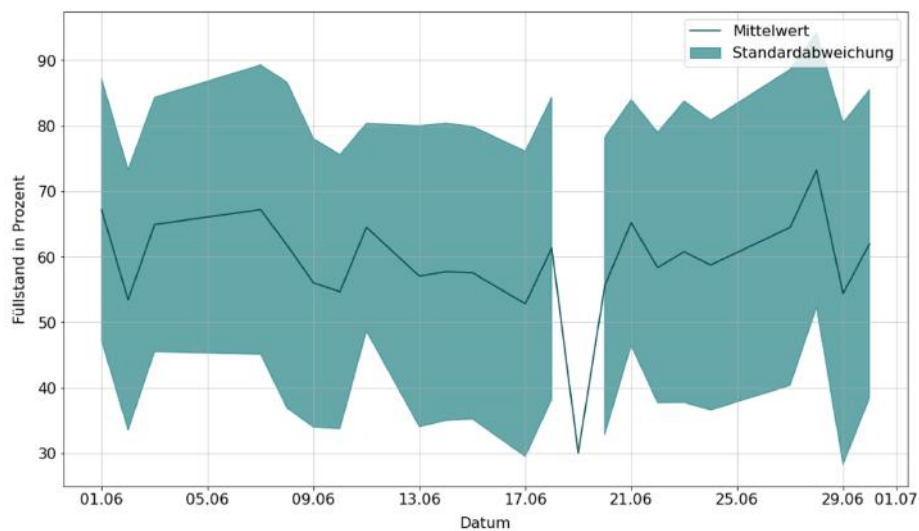
ax.tick_params(labelsize=ticksiz)
ax.set_xlabel("Datum", size=labelsize, labelpad=10)
ax.set_ylabel("Füllstand in Prozent", size=labelsize, labelpad=10)
ax.grid(linewidth=0.5)
ax.xaxis.set_major_formatter(Datumsformat)

ax.legend(prop={'size': legendsize})
plt.subplots_adjust(top=0.9)
fig.suptitle("Durchschnittlicher Füllstand der Container bei Leerung im zweiten
→Quartal", fontsize=titlesize)

plt.show()

```

Durchschnittlicher Füllstand der Container bei Leerung im zweiten Quartal



```
[20]: 1[1["time"]==datetime.date(2022, 6, 19)]
```

```
[20]:
```

	asset_id	name	time	fill_level
218	352bbdb4-229b-11eb-9fa2-7e082caedea2	400252	2022-06-19	30

- Insgesamt sieht man, dass die durchschnittliche Füllstandshöhe relativ konstant bei rund 60 % liegt
- Die hohe Standardabweichung zeigt, dass die Container aber stark unterschiedliche Füllstände bei der Leerung aufweisen
- Hieraus kann man ein recht großes Potenzial zur Optimierung der Glassammlung ableiten
- An einer Stelle (19.06.) verschwindet die Standardabweichung, weil es hier nur einen einzigen Datenpunkt gibt

```
[21]: # Wir nehmen den einen Datenpunkt raus
fs=fs[fs.index!=datetime.date(2022, 6, 19)]

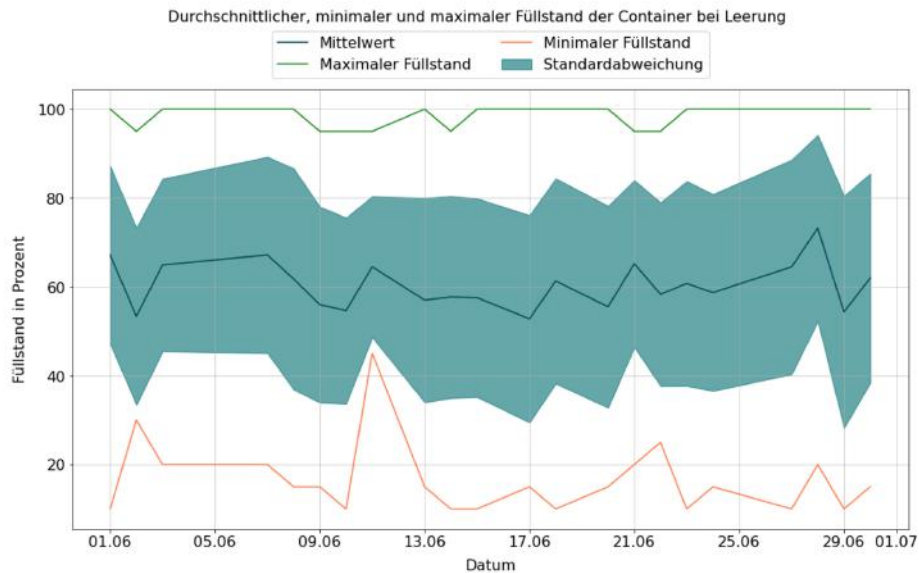
t=fs.index
mean=fs[fs.columns[0]]
std=fs[fs.columns[1]]
#summe=fs[fs.columns[2]]
leerungen=fs[fs.columns[3]]
max_fill=fs[fs.columns[4]]
min_fill=fs[fs.columns[5]]

fig,ax=plt.subplots(figsize=(16,9))
ax.plot(t,mean, c="#004C50", label="Mittelwert")
ax.fill_between(t,np.array(mean)-np.array(std),np.array(mean)+np.array(std),
↳color="#006D72", alpha=0.6, label="Standardabweichung")
ax.plot(t, max_fill, c="forestgreen", label="Maximaler Füllstand")
ax.plot(t, min_fill, c="coral", label="Minimaler Füllstand")

ax.tick_params(labelsize=ticksiz)
ax.set_xlabel("Datum", size=labelsize, labelpad=10)
ax.set_ylabel("Füllstand in Prozent", size=labelsize, labelpad=10)
ax.grid(linewidth=0.5)
ax.xaxis.set_major_formatter(Datumsformat)
ax.legend(loc='upper center', bbox_to_anchor=(0.5, 1.15), ncol=2, prop={'size':
↳legendsize})

plt.subplots_adjust(top=0.85)
fig.suptitle("Durchschnittlicher, minimaler und maximaler Füllstand der
↳Container bei Leerung", fontsize=titlesize)

plt.show()
```



- Hier sieht man zusätzlich, dass es immer Container (d.h. mindestens einen) gibt, die ganz gefüllt sind, aber auch Container, die noch fast leer (ca. 20 % gefüllt) sind.

4 Gefahrene Kilometer

```
[22]: km=pd.read_excel(data_path+"kilometer-glassammelfahrzeuge-RAU.xlsx")
kma=km.loc[km["Kennzeichen"]==224]
kmb=km.loc[km["Kennzeichen"]==1400]
km.head()
#print(min(km["Datum"]), " - ", max(km["Datum"]))
```

```
[22]:
```

	Datum	Kennzeichen	Anfang	Ende	Kilometer
0	2022-04-01	224.0	139970.0	140140.0	170.0
1	2022-04-02	224.0	140140.0	140200.0	60.0
2	2022-04-03	NaN	NaN	NaN	NaN
3	2022-04-04	224.0	140200.0	140342.0	142.0
4	2022-04-05	224.0	140342.0	140449.0	107.0

```
[23]: total=np.sum(km["Kilometer"])
total_a=np.sum(kma["Kilometer"])
total_b=np.sum(kmb["Kilometer"])
print("Die beiden Glassammelfahrzeuge sind im zweiten Quartal insgesamt %d_
↳Kilometer gefahren"%int(total))
```

```
print("Das Fahrzeug mit Kennzeichen BO UB 224 ist %d km gefahren, das Fahrzeug
↳mit Kennzeichen BO UB 1400 ist %d km gefahren."%(int(total_a), int(total_b)))
```

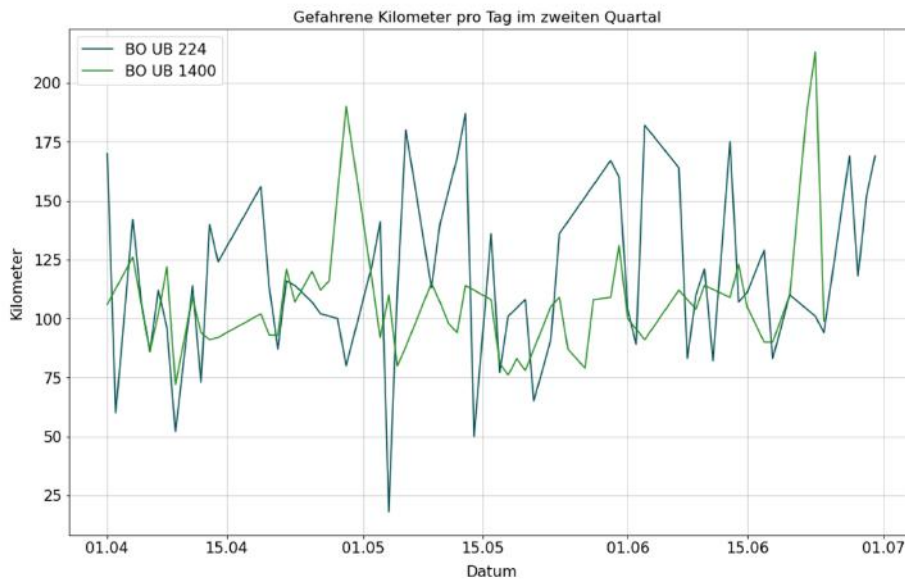
Die beiden Glassammelfahrzeuge sind im zweiten Quartal insgesamt 13060 Kilometer gefahren

Das Fahrzeug mit Kennzeichen BO UB 224 ist 6975 km gefahren, das Fahrzeug mit Kennzeichen BO UB 1400 ist 6085 km gefahren.

```
[24]: # Plote den zeitlichen Verlauf der gefahrenen Kilometer für die beiden
↳Sammelfahrzeuge
fig,ax=plt.subplots(figsize=(16,9))
ax.plot(kma["Datum"], kma["Kilometer"],c="#004C50" , label="BO UB 224")
ax.plot(kmb["Datum"], kmb["Kilometer"],c="forestgreen" , label="BO UB 1400")

ax.tick_params(labelsize=ticksiz)
ax.set_xlabel("Datum", size=labelsize, labelpad=10)
ax.set_ylabel("Kilometer", size=labelsize, labelpad=10)
ax.grid(linewidth=0.5)
ax.xaxis.set_major_formatter(Datumsformat)
ax.legend(prop={'size': legendsize})
plt.subplots_adjust(top=0.95)
fig.suptitle("Gefahrene Kilometer pro Tag im zweiten Quartal",
↳fontsize=titlesize)

plt.show()
```



```
[25]: print("Die beiden Fahrzeuge sind im Durchschnitt %.2f km pro Tag gefahren, die
↳ Standardabweichung beträgt dabei %.2f km"%(np.mean(km["Kilometer"]),np.
↳ std(km["Kilometer"])))
print("Fahrzeug a : mean=%.2f km, std = %.2f km"%(np.mean(kma["Kilometer"]),np.
↳ std(kma["Kilometer"])))
print("Fahrzeug b : mean=%.2f km, std = %.2f km"%(np.mean(kmb["Kilometer"]),np.
↳ std(kmb["Kilometer"])))
```

Die beiden Fahrzeuge sind im Durchschnitt 109.75 km pro Tag gefahren, die Standardabweichung beträgt dabei 34.59 km
 Fahrzeug a : mean=116.25 km, std = 36.24 km
 Fahrzeug b : mean=106.75 km, std = 25.28 km

4.1 Kilometer und Sammelmenge bzw. Anzahl an Leerungen zusammenbringen

```
[26]: # Nehme Tage, an denen nicht gefahren wurde raus
km=km[km["Kilometer"]>0]
kma=kma[kma["Kilometer"]>0]
kmb=kmb[kmb["Kilometer"]>0]

strecke=km.groupby("Datum").agg({"Kilometer":["sum"]})
strecke_a=kma.groupby("Datum").agg({"Kilometer":["sum"]})
strecke_b=kmb.groupby("Datum").agg({"Kilometer":["sum"]})
```

```
[27]: # Daten für Mengen und Kilometer zusammenbringen

menge=pd.DataFrame(summe)
menge.columns=["Menge"]
menge_a=pd.DataFrame(summe_a)
menge_a.columns=["Menge"]
menge_b=pd.DataFrame(summe_b)
menge_b.columns=["Menge"]

strecke=pd.DataFrame(strecke)
strecke.columns=["Kilometer"]
strecke_a=pd.DataFrame(strecke_a)
strecke_a.columns=["Kilometer"]
strecke_b=pd.DataFrame(strecke_b)
strecke_b.columns=["Kilometer"]

data=strecke.join(menge)
data["Menge"]=data["Menge"].fillna(0)
data_a=strecke.join(menge_a)
data_a["Menge"]=data_a["Menge"].fillna(0)
```

```

data_b=strecke.join(menge_b)
data_b["Menge"]=data_b["Menge"].fillna(0)

data.head()

```

```

[27]:           Kilometer  Menge
Datum
2022-04-01         276.0  27.80
2022-04-02          60.0   0.00
2022-04-04         268.0  32.18
2022-04-05         213.0  23.20
2022-04-06         172.0  22.90

```

```

[28]: # Funktion zum Visualisieren der gesammelten Glasmenge und der gefahrenen
↳Strecke

def plot_strecke_und_menge(df, plot_type="lineplot"):
    x=df.index
    y1=df["Kilometer"]
    y2=df["Menge"]
    fig,ax=plt.subplots(figsize=(16,9))
    if plot_type=="lineplot":
        ax.plot(x,y1, c="#004C50", label="Gefahrene Kilometer")
        ax2=ax.twinx()
        ax2.plot(x,y2, c="coral", label="Gesammelte Glasmenge")

        ax.tick_params(labelsize=ticksiz)
        ax.set_xlabel("Datum", size=labelsize, labelpad=10)
        ax.set_ylabel("Gefahrene Kilometer", size=labelsize, labelpad=10)
        ax.grid(linewidth=0.5)
        ax2.tick_params(labelsize=ticksiz)
        ax2.set_ylabel("Gesammelte Glasmenge in Tonnen", size=labelsize,
↳labelpad=10)
        # Ergänze Korrelationskoeffizienten
        ax.text(
            df.index[int(len(list(df.index))/2)], max(y1), "Korrelationskoeffizient:
↳%.4f"%np.corrcoef(y1,y2)[0,1], ha="center", va="center", rotation=0, size=15,
            bbox=dict(boxstyle="round,pad=0.3", fc="white", ec="#004C50", lw=2))

        h1, l1 = ax.get_legend_handles_labels()
        h2, l2 = ax2.get_legend_handles_labels()
        ax.legend(h1+h2, l1+l2, loc=4, prop={'size': legendsize})
        ax.xaxis.set_major_formatter(Datumsformat)

    if plot_type=="scatterplot":
        ax.scatter(y1,y2, label="Datenpunkte")
        ax.tick_params(labelsize=ticksiz)

```



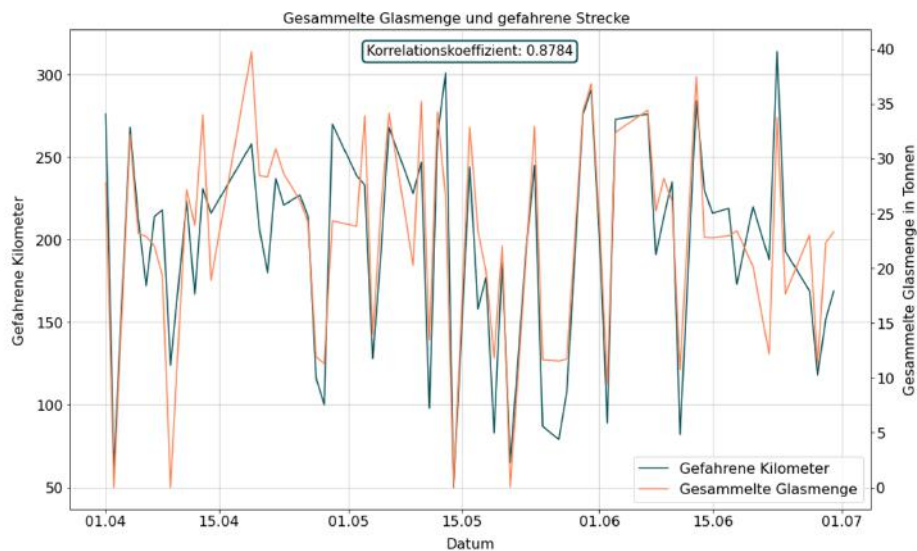
```

ax.set_xlabel("Gefahrene Kilometer", size=labelsize, labelpad=10)
ax.set_ylabel("Gesammelte Menge in Tonnen", size=labelsize, labelpad=10)
ax.grid(linewidth=0.5)
ax.text(
    np.mean(y1), max(y2), "Korrelationskoeffizient: %.4f"%np.
    corrcoef(y1,y2)[0,1], ha="center", va="center", rotation=0, size=15,
    bbox=dict(boxstyle="round,pad=0.3", fc="white", ec="#004C50", lw=2))
m, b = np.polyfit(y1, y2, 1)
X=np.arange(min(y1), max(y1), 0.1)
Y=[m*x+b for x in X]
ax.plot(X,Y, c="coral", lw=2,label="Lineare Regression")
ax.legend(prop={'size': legendsize})

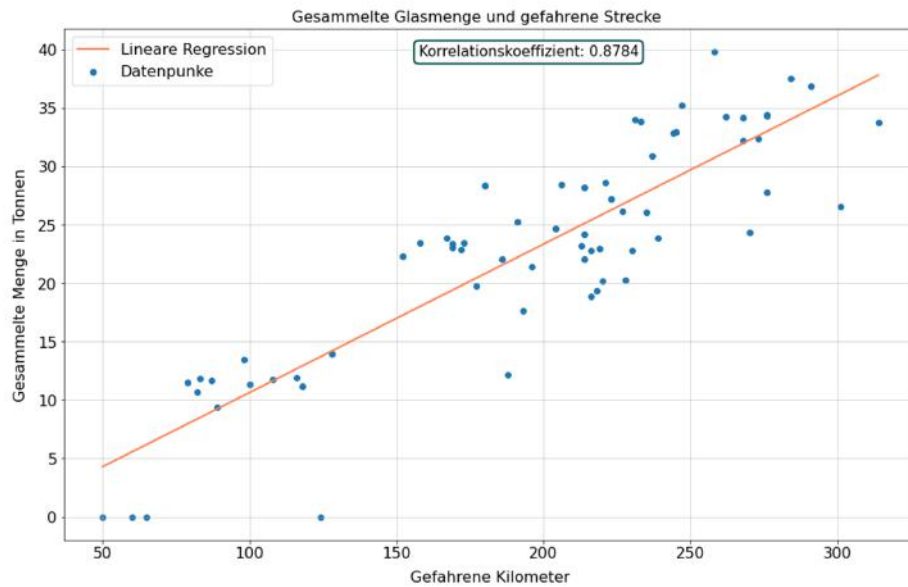
plt.subplots_adjust(top=0.95)
fig.suptitle("Gesammelte Glasmenge und gefahrene Strecke", fontsize=titlesize)
plt.show()

```

```
[29]: plot_strecke_und_menge(data, plot_type="lineplot")
```



```
[30]: plot_strecke_und_menge(data, plot_type="scatterplot")
```



4.2 Gesammelte Glasmenge pro Kilometer bestimmen

```
[31]: # Berechne die gesammelte Glasmenge pro Kilometer
data["Glas pro km"]=data["Menge"]/data["Kilometer"]
data.head()
```

```
[31]:
```

Datum	Kilometer	Menge	Glas pro km
2022-04-01	276.0	27.80	0.100725
2022-04-02	60.0	0.00	0.000000
2022-04-04	268.0	32.18	0.120075
2022-04-05	213.0	23.20	0.108920
2022-04-06	172.0	22.90	0.133140

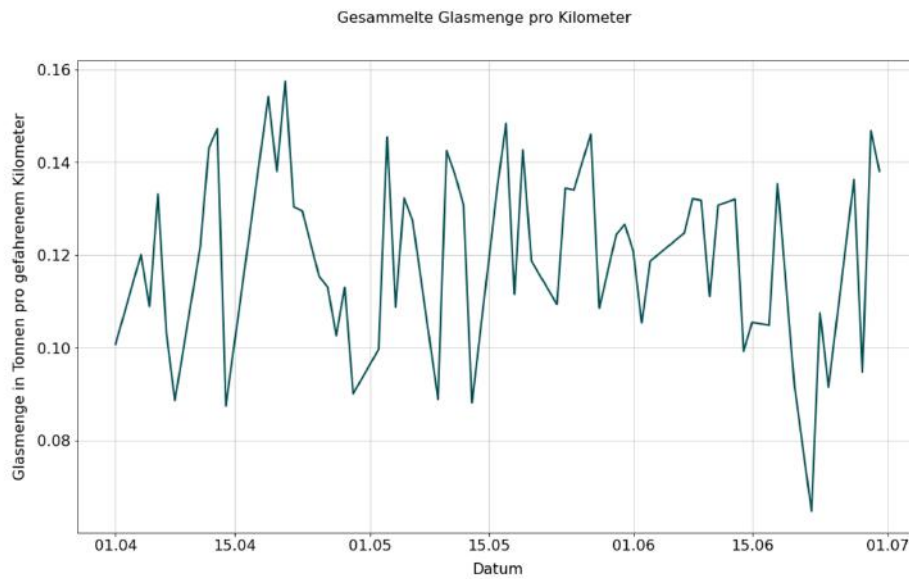
```
[32]: # Plote die gesammelte Glasmenge pro km über die Zeit
fig,ax=plt.subplots(figsize=(16,9))
plotdata=data[data["Glas pro km"]>0] # Nur die Datenpunkte, an denen auch Glas
↳ gesammelt wurde
ax.plot(plotdata.index, plotdata["Glas pro km"], lw=2, c="#004C50",
↳ label="Gesammelte Glasmenge pro Kilometer")
ax.grid(linewidth=0.5)
ax.set_xlabel("Datum", size=labelsizer, labelpad=10)
```

```

ax.set_ylabel("Glasmenge in Tonnen pro gefahrenem Kilometer", size=labelsize,
↳labelpad=10)
ax.tick_params(labelsize=ticksiz)
ax.xaxis.set_major_formatter(Datumsformat)

plt.subplots_adjust(top=0.9)
fig.suptitle("Gesammelte Glasmenge pro Kilometer", fontsize=titlesize)
plt.show()

```



```

[33]: print("Im Durchschnitt wurden %.3f t Glas pro km gesammelt, die
↳Standardabweichung betragt dabei %.3f t/km"%(np.mean(data["Glas pro km"]),np.
↳std(data["Glas pro km"])))

```

Im Durchschnitt wurden 0.113 t Glas pro km gesammelt, die Standardabweichung betragt dabei 0.034 t/km

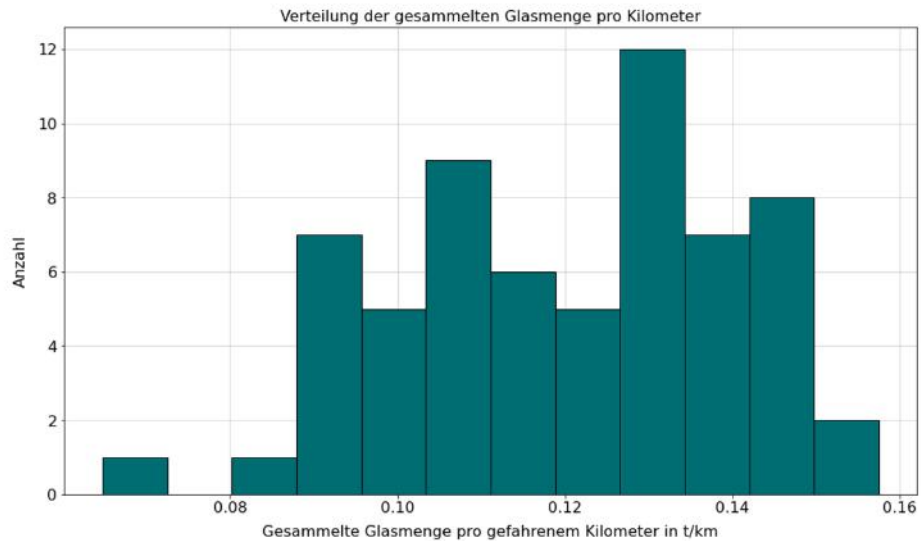
```

[34]: fig,ax=plt.subplots(figsize=(16,9))

plotdata.hist(ax=ax, column="Glas pro km", color="#006D72", ec="black", bins=12)
ax.grid(linewidth=0.5)
ax.set_ylabel("Anzahl", size=labelsize, labelpad=10)
ax.set_xlabel("Gesammelte Glasmenge pro gefahrenem Kilometer in t/km",
↳size=labelsize, labelpad=10)
#ax.set_xticks(range(0,110,10))

```

```
ax.tick_params(labelsize=ticksizel)
ax.set_axisbelow(True)
plt.title("Verteilung der gesammelten Glasmenge pro Kilometer",
fontsize=titlesize);
plt.show()
```



5 Export als PDF

```
[35]: %%capture
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!pip install py pandoc
```

```
[36]: %%capture
!jupyter nbconvert --to PDF "/content/drive/Shared drives/Datenexploration_
Zolitron/Datenexploration.ipynb"
```