



Mikołaj Nowak

Wykorzystanie modelu grawitacji w logistyce

The use of the gravity model in logistics

Praca magisterska

Promotor: dr hab. Marcin Szymkowiak, prof. UEP

Data przyjęcia:

Podpis promotora

Kierunek: Informatyka i ekonometria

Specjalizacja:

Poznań 2021

Spis treści

Wstęp	3
1 Współrzędne geograficzne w logistyce	4
1.1 Czym jest geokodowanie i gdzie jest wykorzystywane	4
1.2 Wykorzystanie geokoordynat w logistyce	5
1.3 Źródła i aplikacje wykorzystane do wyznaczenia geokoordynat	6
1.3.1 eMapa	6
1.3.2 GeoNames	9
1.3.3 PTV	9
1.3.4 Google	10
1.3.5 Nominatim	10
1.4 Wykorzystanie formuły Haversina	11
1.5 Wizualizacja oraz interpretacja wygenerowanych danych	13
1.6 Utworzenie par adresowych	15
1.6.1 Obliczenie tras	16
2 Definicja i zastosowanie modelu grawitacyjnego	17
2.1 Definicja modelu grawitacyjnego	18
2.2 Grawitacyjna analiza handlu w Pythonie	20
3 Estymacja modelu grawitacji w analizie emisji spalin samochodów ciężarowych	23
3.1 Analiza parametrów modelu grawitacji	23
3.2 Analiza porównawcza parametrów modelu ciążenia	27
3.2.1 Zużycie paliwa	30
3.2.2 Amoniak	33

3.2.3	Benzen	36
3.2.4	Tlenek węgla	39
3.2.5	Dwutlenek węgla	42
3.2.6	Węglowodory	45
3.2.7	Metan	48
3.2.8	Tlenki azotu	51
3.2.9	Podtlenek azotu	54
3.2.10	Cząstki stałe	57
3.2.11	Dwutlenek siarki	60
	Zakończenie	64
	Spis tabel	67
	Spis rysunków	69
	Kody języka Python	70

Wstęp

Troska o środowisko naturalne często przestaje być modą, a staje się jednym z codziennych działań podejmowanych przez korporacje, dla których dobro naszej planety jest wielką troską. Dbłość o środowisko naturalne jest coraz częściej uwzględniane w głównych założeniach społecznej odpowiedzialności. Najbardziej oczywistym rozwiązaniem jest ograniczenie transportu drogowego lub wprowadzenie najnowszych technologii niwelujących emisyjność gazów do atmosfery. Jednakże wraz z postępującym rozwojem rynku oraz zwiększenia jego dynamiki, ograniczenie transportu drogowego byłoby niekorzystne dla wszystkich. W konsekwencji ucierpiałyby potrzeby konsumentów, interesy przedsiębiorstw, a także wyniki gospodarcze poszczególnych krajów. Na wprowadzenie najnowszych rozwiązań czy zastąpienie starego, pocziwego diesla paliwami alternatywnymi, jest czymś na co nie każdy przedsiębiorca może sobie pozwolić. Jednakże jest możliwe - a nawet konieczne, w obliczu postępujących zmian klimatycznych - aby uczynić transport tak zrównoważonym, jak to tylko możliwe.

Badania naukowe wykazują powolny lecz nieustanny wzrost średniej temperatury Ziemi. Zjawisko to wywołane jest, między innymi, zwiększoną emisją gazów cieplarnianych. Pojazdy ciężarowe odpowiadają za 22% emisji pochodzących z transportu drogowego w UE, jednocześnie stanowią niecałe 5% floty¹. Coraz więcej osób i instytucji zdaje sobie sprawę z wpływu, jaki branża TSL (Transport Spedycja Logistyka) (Bartosik i in., 2015) odciska na środowisku naturalnym i zależy im na tym, aby jak najbardziej ten „ślad” minimalizować.

Celem pracy jest szerokie przeanalizowanie sytuacji emisji spalin i poszczególnych pierwiastków chemicznych w poszczególnych regionach w Polsce. Dane potrzebne do przeprowadzenia takiej analizy są bardzo specyficzne i trudne do otrzymania w ogólnie-

¹<https://leonardo-energy.pl/artykuly/transport-ciezarowy-i-zmniejszenie-emisji-spalin-dlaczego-si-e-to-oplaci-10-faktow/#:~:text=Normy%20efektywno%C5%9Bci%20paliwowej%20dla%20samocho%C3%B3w,w%C4%99gla%20rocznie%20do%202030%20r.>

dostępnych bazach danych, dlatego na potrzeby poniższej pracy, zostały własnoręcznie wygenerowane potrzebne dane. Do tego celu wykorzystano wiele narzędzi geokodujących, bibliotek pythonowych, rozwiązań relacyjnych baz danych, serwisów API oraz usług wyznaczających trasy. Same prace przygotowawcze, zbieranie danych, ich weryfikacja oraz transformacja zajęły większość czasu poświęconego owemu badaniu. W niniejszej pracy podjęto próbę wykorzystania modelu grawitacji do oszacowania ilości wytwarzanych emisji, podczas ruchu drogowego samochodów ciężarowych w Polsce. Obiektami przestrzennymi, dla których prowadzona była analiza były województwa w Polsce.

W pierwszym rozdziale opisane zostały źródła oraz metodologia wykorzystana do pozyskania niezbędnych danych terytorialnych w Polsce. Wyzwaniem było pozyskanie wszystkich aktualnych kodów pocztowych w Polsce. W tym celu wykorzystano wiele darmowych oraz prywatnych baz danych. Aby mieć możliwość dalszej pracy na kodach pocztowych konieczne było przedstawienie ich w postaci współrzędnych geograficznych. Do tego zostało użytych wiele serwisów i usług internetowych oferujących geokodowanie adresów. Dzięki temu, mając kody pocztowe przedstawione w układzie współrzędnych geograficznych, możemy je łatwo wykorzystywać i wizualizować w dalszych analizach. Na podstawie otrzymanych geokoordynat zostały wygenerowane losowe trasy dla każdego kodu pocztowego, a następnie obliczone rzeczywiste parametry pokonania tych tras według narzędzia oferowanego przez firmę PTV.

Moduł PTV x Route umożliwia wyznaczenie rzeczywistych tras pomiędzy punktami, z uwzględnieniem parametrów samochodu ciężarowego. Dzięki temu uwzględniane są prędkości, ograniczenia oraz opłaty drogowe dla tirów. Również zwracane są informacje ile spalin generuje pojazd przy pokonaniu danej trasy. To właśnie te emisje poszczególnych pierwiastków, które dostajemy w odpowiedzi od serwisu PTV, będą kluczowe dla całego badania. Trasy przeliczały się jedna po drugiej, przez co estymacja 943036 tras zajęła ponad tydzień ciągłego działania sprzętu. Był to bardzo czasochłonny proces, który wymagał ciągłej kontroli, aby jakość danych była jak najlepsza.

W kolejnym rozdziale przedstawione zostały teoretyczne podstawy związane z metodą użytą w pracy jaką są modele grawitacyjne (ciężenia). Przytoczona została ich definicja, postępowanie badawcze, rozwój i wykorzystywane oprogramowanie. Ważnym elementem było także przedstawienie postępowania z danymi języku programowania

Python, który zostanie użyty do obliczeń w pracy.

W trzecim, finalnym rozdziale przedstawione zostały wygenerowane w pracy modele grawitacyjne oraz wykorzystane do tego celu biblioteki języka Python potrzebne do analizy. Zostały także przedstawione i opisane najważniejsze wyniki uzyskane w pracy, czyli analiza opisowa oraz wizualizacje w formie kartogramów.

Niniejsza praca przedstawia tematykę bardzo ciekawą z perspektywy każdego człowieka, dla którego dbałość o środowisko jest nieodłączną częścią życia. Wyniki badań są bardzo ciekawe i przedstawiają różne perspektywy emisyjności poszczególnych pierwiastków. Daje to również szerokie pole do dalszych analiz oraz możliwość znalezienia nowych rozwiązań związanych z poprawą jakości powietrza i środowiska.

Rozdział 1

Współrzędne geograficzne w logistyce

1.1 Czym jest geokodowanie i gdzie jest wykorzystywane

Większość systemów nawigacyjnych używanych w transporcie lądowym wykorzystuje współrzędne geograficzne. Chociaż wydają się, że korzystają one wyłącznie z adresów i nazw ulic w postaci zwykłego tekstu, w rzeczywistości konwertują one wskazane adresy na szerokość oraz długość geograficzną. Bazując na otrzymanych współrzędnych określają lokalizację oraz wyznaczają najbardziej optymalne na dany moment trasy. Proces konwersji adresu na odpowiadającą mu szerokość oraz długość geograficzną nazywany jest geokodowaniem (geocoding). Określenie współrzędnych geograficznych wybranych punktów terenowych sprowadza się na nich do wyznaczenia szerokości i długości geograficznej (Borowczyk i Kaczmarek, 2007). Przykładową reprezentacją adresu „Towarowa 55, 60-995 Poznań” w układzie współrzędnych geograficznych będzie : **52.4063426319362°N, 16.916891439988806°E**¹. Odwrotnym procesem do geokodowania jest geokodowanie wsteczne (odwrócone), dla którego wejściem są współrzędne geograficzne, a wynikiem zaś adres. Dane powstałe w wyniku geokodowania są wykorzystywane w systemach informacji geograficznej (Geographic Information System (GIS))² oraz systemach nawigacyjnych. System GIS umożliwia gromadzenie, zarządzanie, analizowanie oraz wizualizację danych na mapie w oparciu o ich lokalizację geograficzną (Ali, 2020). Wykorzystuje on dane przestrzenne do tworzenia map i mo-

¹<https://www.google.pl/maps?hl=en&tab=wl>

²<https://www.omnisci.com/technical-glossary/gis>

deli 3D, pokazując wzorce oraz relacje występujące w danych. Firmy wykorzystują systemy GIS do wybierania lokalizacji atrakcyjnych biznesowo (nowe lokalizacja sklepu stacjonarnego), zarządzania łańcuchem dostaw, profilowania swoich klientów czy też targetowania kampanii reklamowych (Huisman i Rolf, 2009). Rodzaje geokodów możemy podzielić ze względu na³:

- Sposób kodowania - liczbowe, literowe, mieszane,
- Terytorium jakie obejmują - lądy, wody, kontynent, państwo, miasto, itp.,
- Rodzaj kodowanych obiektów - państwa, regiony administracyjne, miasta, lotniska, itp.

Z wykorzystania współrzędnych geograficznych wynika wiele pozytywów. Jest to format, z którym każdy z nas, lub przynajmniej większość, miało do czynienia, wie jak go interpretować oraz korzystać z niego w praktyce. Notacja ta może odwoływać się do bardzo małego i specyficznego punktu na Ziemi, dzięki czemu możemy znacznie zwiększyć dokładność naszych geokodowań. Tak jak adresy na całym świecie różnią się formatem w jakim są przedstawiane w danym regionie, to reprezentacja szerokości oraz długości geograficznej jest jedna i taka sama wszędzie.

1.2 Wykorzystanie geokoordynat w logistyce

Jak podaje autor książki „Wprowadzenie do logistyki” (Słowiński, 2008), logistyka to „*pewna koncepcja zarządzania procesami przepływu dóbr, oparta na zintegrowanym i systemowym ujmowaniu tych procesów. Ideą zarządzania logistyką jest koordynacja przepływów w celu minimalizacji kosztów tych przepływów*”. Aby zapewnić jak najlepszą koordynację oraz poniesienie przy tym jak najmniejszych kosztów, konieczne jest przemyślane zaplanowanie oraz sprawne przetransportowanie wcześniej wspomnianych przepływów. Nie będzie to wykonalne, gdy nie będziemy w posiadaniu dokładnych danych lokalizacyjnych miejsc załadunku oraz rozładunku. Według artykułu amerykańskiej firmy tworzącej aplikacje do sprawnego zarządzania i monitorowania ruchu ciężarówek DOCK41⁴, koszty wynikające z niedokładności adresów klientów końcowych sięgają milionów dolarów rocznie. Adresy i ich wirtualne reprezentacje na mapach w aplikacjach

³<https://www.szkolnictwo.pl/szukaj,Geokodowanie>

⁴<https://dock411.com/using-street-addresses-cost-trucking-and-logistics-millions-per-year/>

nawigacyjnych działają wystarczająco dobrze dla mniejszych pojazdów. Jednak nie-
rzadko zdarzają się sytuację, gdzie wynik dla danego adresu jest błędny. Sytuacja jest
o wiele bardziej skomplikowana, gdy nie jedziemy samochodem osobowym, lecz pojaz-
dem ciężarowym. Nawrócenie dwudziestometrowym pojazdem na drodze publicznej
jest praktycznie niemożliwe. Biorąc pod uwagę wszystkie utrudnienia i niedogodności
wynikające z faktu prowadzenia tak dużych pojazdów, nie można sobie pozwolić na
błędy w systemach geolokalizacyjnych. Na potrzeby dalszych badań, zebrane oraz zge-
okodowane z wykorzystaniem wielu różnych narzędzi wszystkie kody pocztowe w Pol-
sce.

1.3 Źródła i aplikacje wykorzystane do wyznaczenia geokoordynat

Aby zminimalizować prawdopodobieństwo błędnej geolokalizacji dla kodów pocztowych, posłużymy się wieloma serwisami do geolokalizacji. Dzięki temu nasze dane, na których będziemy opierać dalsze badania z wykorzystaniem ciążenia, będą jak najdokładniejsze oraz pozbędziemy się ryzyka przekłamania wyników, ze względu na jego złe dopasowanie w układzie współrzędnych.

Adresy wykorzystywane do geolokalizacji pochodzą z ogólnodostępnej i darmowej bazy kodów pocztowych z serwisu www.kody-pocztowe.dokladnie.com⁵. Baza została poszerzona o brakujące w niej adresy z takich serwisów jak GeoNames, prywatnej bazy eMapa oraz ogólnodostępnych baz danych, w celu budowy jak najbardziej kompletnego zbioru kodów pocztowych na terenie Polski. Posiadając referencyjną bazę kodów pocztowych, jesteśmy w stanie tworzyć zapytania do zewnętrznych aplikacji oraz filtrować dostępne zasoby w celu uzyskania odpowiednich kodów pocztowych w oparciu o bazę referencyjną.

1.3.1 eMapa

EMapa jest jednym z prywatnych zasobów grupy Raben. Jest to jedna z największych firm logistycznych w Polsce, która swoją działalność w RP rozpoczęła w 1991 r.⁶. Kody


⁵<https://www.kody-pocztowe.dokladnie.com/>

⁶https://pl.wikipedia.org/wiki/Raben_Group

pocztowe pochodzą z bazy adresów wykorzystywanych przy świadczeniu usług logistycznych dla swoich klientów. Jako, że jest to dynamiczna baza danych oraz jej kompletność zależy od zakresu świadczenia usług grupy Raben, możemy spodziewać się tu wielu braków danych w ujęciu całej Polski, natomiast bardzo dużej dokładności wygenerowanych dla dostępnych adresów geokoordinat.

Dostępne w zbiorze adresy są definiowane z dokładnością do nazwy miasta. Jako że interesujące są dla nas tylko unikalne wartości kombinacji kod pocztowy-miasto, a w zbiorze występują powielające się geokoordinaty dla danych miast, to wymagane jest ujednolicenie danych. W tym celu, z adresów wyliczone zostały scentralizowane, uśrednione współrzędne geograficzne, tak aby dla danego miasta i kodu pocztowego występował tylko i wyłącznie jeden rekord. Dzięki temu, mając wiele rekordów odnoszących się do jednego miasta, otrzymujemy jedną wartość, będącą środkiem wszystkich rekordów w układzie współrzędnych geograficznych. W celu dokładniejszego zrozumienia, idea ta zaprezentowana jest na rysunku numer 1.1.

PL	RAWICZ	63-900	16.86889267	51.61479950
PL	RAWICZ	63-900	16.85941696	51.61501312
PL	RAWICZ	63-900	16.85245132	51.60952759
PL	RAWICZ	63-900	16.85245132	51.60952759
PL	RAWICZ	63-900	16.83873749	51.61207962



PL	RAWICZ	63-900	16.85438995	51.61218948
----	--------	--------	-------------	-------------

Rysunek 1.1. Wyznaczanie środka punktów

Źródło: Opracowanie własne na podstawie zebranych danych

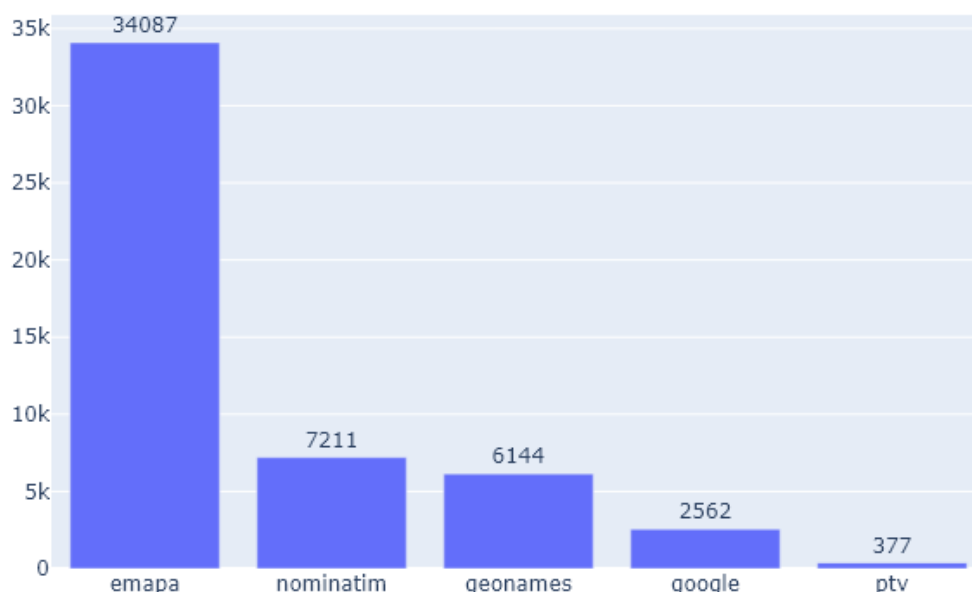
Jako że wyznaczanie średniej szerokości oraz długości geograficznej odbywa się na układzie współrzędnym dwóch zmiennych, to do ich wyznaczenia możemy wykorzystać następujący wzór ⁷:

$$\frac{\sum \text{Długość geograficzna}}{\sum n}; \frac{\sum \text{Szerokość geograficzna}}{\sum n}, \quad (1.1)$$

gdzie n jest liczbą rekordów dla danego adresu.

⁷<https://zpe.gov.pl/a/dlugosc-odcinka-srodek-odcinka/DKW85YFr9>

Niestety pomimo względnie wysokiej dokładności geokoordynat w bazie eMapa, liczba unikalnych adresów już nie jest tak satysfakcjonująca. Na rysunku 1.2 możemy zobaczyć jak dużo braków występuje we wspomnianym wcześniej zbiorze danych, względem naszej referencyjnej bazy kodów i miast.



Rysunek 1.2. Liczba brakujących adresów w poszczególnych źródłach danych

Źródło: Opracowanie własne w programie Python na podstawie zebranych danych

W danych występują aż 34087 braki, co stanowi 57.82% całego zbioru. Wynik ten jest spowodowany tym, że w bazie znajdują się wyłącznie adresy, które były wykorzystywane do świadczenia usług grupy Raben. Gdybyśmy opierali się w swoich badaniach wyłącznie o dane z eMapy, byłby to wynik nie do przyjęcia. Zbyt wiele braków danych nie może nam zapewnić relatywnie dobrych wyników. Jako że korzystamy z wielu źródeł danych wejściowych, to warto wykorzystać dostępne w eMapie rekordy, które pomogą nam w dokładniejszej estymacji centrów punktów w układzie współrzędnych geograficznych.

1.3.2 GeoNames

Geograficzna baza danych GeoNames jest powszechnie dostępna w sieci i składa się z ponad 25 milionów nazw geograficznych. Dane są bezpłatnie dostępne poprzez szereg serwisów internetowych oraz podstawowe eksporty do plików csv. GeoNames integruje dane geograficzne takie jak nazwy adresów, populacje, podział terytorialny itd. Wszystkie wartości szerokości oraz długości geograficznych prezentowane są w formacie WGS84 (World Geodetic System 1984). Użytkownicy mogą ręcznie edytować, dodawać i poprawiać istniejące w bazie nazwy za pomocą przyjaznego interfejsu wiki⁸. Baza GeoNames ma znacznie mniej braków w adresach od eMapy. 6144 niepełnych rekordów jest o wiele lepszym wynikiem od poprzedniej bazy, natomiast stanowi to ponad 10.4 % całości bazy co również może nie być zadowalającym wynikiem.

1.3.3 PTV

PTV Geocoding API jest komponentem, dostępnym jako usługa sieciowa w chmurze lub jako rozwiązanie on-premise. Dzięki swojemu standardowemu interfejsowi komponent ten można łatwo zintegrować z istniejącą architekturą oprogramowania. Szybko realizuje on żądania geokodowania - zarówno dla pojedynczych lokalizacji, jak i dla całych baz danych. PTV oferuje maksymalną elastyczność i najwyższą precyzję, dzięki czemu można wprowadzać informacje adresowe bez szczególnych wymagań formatowania. Udostępniana jest bardzo precyzyjna baza danych, umożliwiająca perfekcyjne dopasowanie współrzędnych geograficznych i geolokalizacji⁹. Dodatkowa walidacja adresów sprawdza i ocenia jakość dopasowania danych wejściowych i wyjściowych, aby pomóc w wyorze jak najlepszego adresu. W ten sposób możemy być pewni, że zawsze korzystamy z aktualnych danych oraz unikamy nieefektywności¹⁰. Patrząc ponownie na rysunek 1.2 przedstawiający braki w adresach możemy zauważyć, że PTV jest tutaj na czele, jeśli chodzi o kompletność danych. Zdefiniowano tutaj 377 braków, które stanowią 0.64% względem całej bazy danych. Jest to najlepszy wynik wśród wszystkich źródeł danych branych pod uwagę.

⁸<https://www.geonames.org/about.html>

⁹https://xserver2-test.cloud.ptvgroup.com/dashboard/Content/UseCases/Geocoding/UC_Searching_an_address.html

¹⁰<https://www.ptvgroup.com/en/solutions/products/ptv-xserver/developer-zone/geocoding-api/>

1.3.4 Google

API od GOOGLE'a zapewnia bezpośredni dostęp do danych geolokalizacyjnych poprzez żądania HTTP. Usługa ta jest przeznaczona do geokodowania statycznych adresów, w celu ich wizualizacji na mapie lub w innych aplikacjach¹¹. Zapytania o współrzędne geograficzne poszczególnych miast były zbudowane w następujący sposób:
`https://maps.googleapis.com/maps/api/geocode/json?address={kod_pocztowy}+{miasto}®ion=pl&key={API_KEY}`

Parametr o nazwie „address” składa się w tym wypadku z dwóch składowych. Jednym jest kod pocztowy naszego adresu, a drugim jest miasto. Żeby mieć pewność słuszności naszego zapytania oraz, że dostaniemy wynik wyłącznie dla Polski, został dodany jeszcze jeden parametr „region”, który zapewnia, że zwrócone geokoordynaty leżą w obrębie RP. Google znajduje się na drugim miejscu jeśli chodzi o kompletność danych. Zdefiniowano tutaj nieco ponad 2500 braków, co w ujęciu procentowym stanowi 4,35%. Patrząc na wszystkie inne źródła z wyjątkiem PTV, jest to bardzo dobry wynik.

1.3.5 Nominatim

Nominatim to narzędzia do wyszukiwania danych w zakresie OpenStreetMap (OSM) według nazw adresów oraz do generowania syntetycznych adresów punktów OSM w formie geokodowania odwrotnego (reverse geocoding)¹². Projekt OpenStreetMap tworzy i rozpowszechnia darmowe dane geograficzne dla całego świata¹³. API do wyszukiwania współrzędnych geograficznych poszczególnych punktów, pozwala na ich wyszukiwanie na podstawie nazw adresów lub ich opisów. Nominatim obsługuje zarówno predefiniowane zapytania strukturalne, jak i dowolnie zbudowane zapytania¹⁴. Patrząc na kompletność danych dla usługi Nominatim, możemy zauważyć, że jest ono na przedostatnim miejscu. Od GeoNames różni go ponad 1000 niekompletnych adresów.

¹¹<https://developers.google.com/maps/documentation/geocoding/overview>

¹²<https://nominatim.org/release-docs/develop/>

¹³https://wiki.openstreetmap.org/wiki/Main_Page/

¹⁴<https://nominatim.org/release-docs/develop/api/Search/>

1.4 Wykorzystanie formuły Haversina

Z wykorzystaniem wcześniej wspomnianych narzędzi i zbiorów danych, zostały wygenerowane i przypisane do adresów współrzędne geograficzne, których łącznie unikalnych jest 58951. Korzystając z wielu źródeł, do wyznaczenia pojedynczego punktu, trudno stwierdzić, które z nich jest najdokładniejsze w swoich wyliczeniach i z którego powinniśmy korzystać w dalszych pracach. W celu otrzymania jak najbardziej dokładnych danych, na wygenerowanych wcześniej środkach punktów w układzie współrzędnych geograficznych, została zastosowana formuła Haversina, która może określić odległość między dwoma obiektami na powierzchni kuli (Prasetya i in., 2019). Wzór Haversina jest bardzo ważnym równaniem w nawigacji, zwracającym rzeczywiste odległości wielkiego okręgu między dwoma punktami na sferze, bazując na ich długości i szerokości geograficznej¹⁵. Wyznacza on rzeczywiste odległości pomiędzy dwoma punktami w linii prostej, z wykluczeniem różnic terenu (gór itp.). Jako referencja, został przyjęty wzór z portalu <https://www.movable-type.co.uk/scripts/latlong.html>:

$$a = \sin(\Delta\phi/2) + \cos\phi_1 * \cos\phi_2 * \sin(\Delta\lambda/2)$$
$$c = 2 * \operatorname{atan2}(\sqrt{a}, \sqrt{1-a})$$
$$d = R * c,$$

gdzie ϕ jest szerokością geograficzną, λ jest długością geograficzną, a R promieniem kuli ziemskiej (6 371 km).

Aby obliczyć te wartości na zbiorze danych, została utworzona funkcja "haversine_formula". Jako parametry wejściowe funkcji, przyjęte zostały punkty geograficzne poszczególnych miejsc:

lokalizacja = (szerokość geograficzna, długość geograficzna)

Utworzona funkcja, prezentuje się w następujący sposób:

```
def haversine_formula(location_1, location_2):  
    lat1 = location_1[0]  
    lon1 = location_1[1]  
    lat2 = location_2[0]  
    lon2 = location_2[1]  
  
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])  
  
    dlon = lon2 - lon1  
    dlat = lat2 - lat1
```

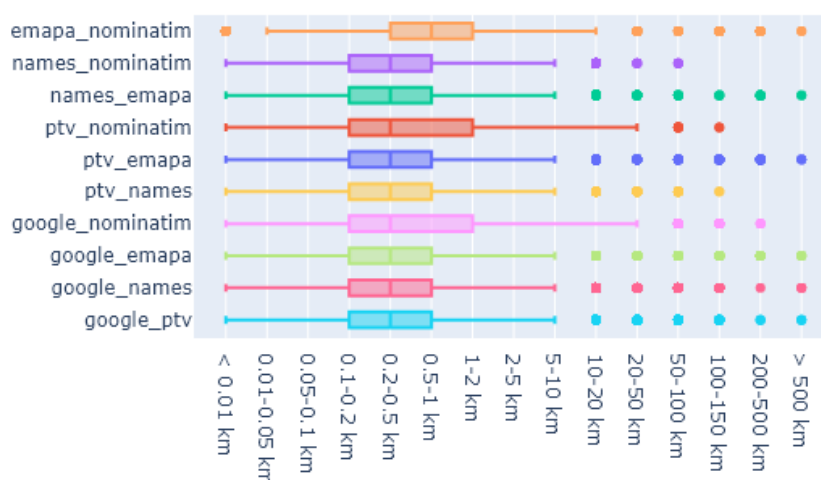
¹⁵https://rosettacode.org/wiki/Haversine_formula

```
a~= sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2      13
c = 2 * asin(sqrt(a))                                              14
R = 6371                                                            15
                                                                    16
result_in_meters = R * c * 1000                                     17
                                                                    18
return result_in_meters                                           19
```

Aby funkcja działała prawidłowo, niezbędna jest zamiana wartości długości i szerokości geograficznej na radiany, co otrzymujemy poprzez zastosowanie funkcji `map()` na naszych zmiennych. Wykorzystując wyżej pokazaną funkcję, zostały obliczone względne odległości pomiędzy każdym ze źródeł współrzędnych geograficznych. Trudno stwierdzić, czy dana wartość jest prawidłowym dopasowaniem dla punktu bez porównania z innymi danymi, więc porównanie wielu wartości względem siebie pomoże w identyfikacji błędnych estymacji oraz wykluczeniu ich ze zbioru. Pozwala nam to wyeliminować dane, które są niewłaściwie dopasowane i znacząco odbiegają od pozostałych geokoordynat.

1.5 Wizualizacja oraz interpretacja wygenerowanych danych

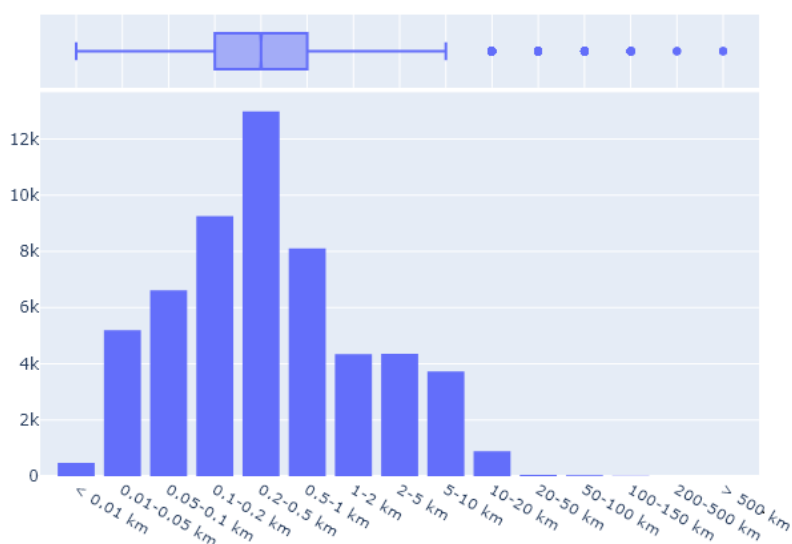
Posiadanych danych nie możemy zostawić bez jakiegokolwiek interpretacji. W tym celu zostały wygenerowane różnych rodzajów wykresy, aby jak najbardziej zrozumieć nasze dane. Jako pierwszy, utworzony został wykres pudełkowy, pokazujący różnicę w odległościach, wyliczonych wcześniej wspomnianą funkcją wykorzystującą formułę Haversina.



Rysunek 1.3. Porównanie źródeł geokoordynat

Źródło: Opracowanie własne w programie Python na podstawie zebranych danych

Jak możemy zauważyć na powyższym wykresie, większość źródeł różni się od siebie w podobny sposób. Jedynie różnice w relacjach emapa-nominatim, ptv-nominatim oraz google-nominatim są na wyższym poziomie niż wszystkie inne. Może to oznaczać, że dane otrzymane z serwisu nominatim, są mniej dokładne od pozostałych na co wskazują duże różnice względem innych wykorzystywanych źródeł. Najbardziej przybliżone wyniki do nominatim posiada serwis geonames. W relacji names-nominatim nie ma rekordów o różnicy większej niż 100km, a najczęściej występujące różnice są w przedziale 0.1km - 1km co wskazuje na wysoką korelację pomiędzy tymi dwoma serwisami. Najbardziej odbiegającymi od nominatim, wydają się być google oraz ptv. Patrząc na wykres, różnice pomiędzy relacjami ptv-nominatim oraz google-nominatim są największe, co może wskazywać na to, że przeciwieństwem najmniej dokładnego nominatim, są w tym wypadku google oraz serwis ptv. Dokładne różnice w odległości przedstawione w metrach pomiędzy google i ptv ukazuje rysunek 1.4:

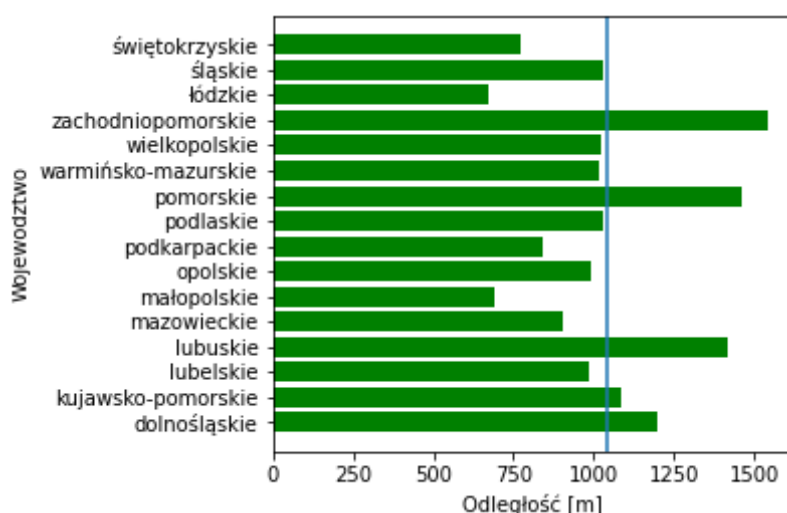


Rysunek 1.4. Różnice odległości relacji Google-PTV

Źródło: Opracowanie własne w programie Python na podstawie zebranych danych

Najwięcej adresów różni się między sobą od 0.2 km do 0.5 km. Nie jest to duża różnica, biorąc pod uwagę, że wszystkie nasze punkty są determinowane z dokładnością do miasta, a nie ulicy czy numeru domu. Z tego faktu, środek miasta może być inaczej definiowany przez każde z wykorzystywanych źródeł danych.

Kolejnym typem wykresu jest wykres słupkowy, pokazujący średnią różnicę pomiędzy scentralizowanymi punktami współrzędnych geograficznych w danych województwach. Przedstawiony on jest na rysunku 1.5, który znajduje się poniżej.



Rysunek 1.5. Średnia odległość pomiędzy punktami w danym województwie

Źródło: Opracowanie własne w programie Python na podstawie zebranych danych

Wyszczególnione zostały wszystkie województwa oraz odpowiadające im średnie.

Pionową, niebieską kreską została naniesiona ogólna, średnia wartość odległości pomiędzy dwoma punktami, która oszacowana została na wartość 1041 m. Województwem o najmniejszej różnicy odległości okazało się województwo łódzkie, gdzie współrzędne geograficzne różniły się od siebie średnio o 669.48 m. Kolejnym pod tym względem województwem, było województwo małopolskie, z wynikiem 688.11 m oraz województwo świętokrzyskie - 770.45m. Możemy wywnioskować, że geokoordynaty dla adresów znajdujących się w województwie łódzkim, są najdokładniejsze względem innych województw w naszym zbiorze danych. Najmniej dokładnym województwem jest województwo zachodniopomorskie, w którym punkty, różniły się między sobą średnio o aż 1544.34 m. Następnie było to województwo pomorskie - 1458.48 m oraz lubuskie - 1417.28 m.

1.6 Utworzenie par adresowych

Następnym krokiem, było wygenerowanie tras pomiędzy dostępnymi adresami i odpowiadającymi im koordynatami. Schemat tworzenia tras jest następujący. Dla każdego adresu w bazie, zostało przypisanych 16 innych adresów, gdzie każdy z nich pochodził z innego województwa. Dzięki temu, dla danego rekordu, zostało wygenerowanych 16 unikalnych tras pomiędzy miastami z każdego województwa w Polsce. Przykładowo, dla adresu 00-002, Warszawa, zostało przyporządkowanych 16 adresów z całej Polski, gdzie z każdego województwa wylosowany został tylko jeden. Przykład wygenerowanych tras dla kodu pocztowego 00-002 w Warszawie znajduje się w poniższej tabeli 1.1.

Tabela 1.1. Trasy wygenerowane dla 00-002, Warszawa

kod z	miasto z	województwo z	kod do	miasto do	województwo do
00-002	Warszawa	mazowieckie	96-321	Słubica Dobra	mazowieckie
00-002	Warszawa	mazowieckie	59-160	Nowa Kuźnia	dolnośląskie
00-002	Warszawa	mazowieckie	98-358	Tuchań	łódzkie
00-002	Warszawa	mazowieckie	87-640	Zimny Zdrój	kujawsko-pomorskie
00-002	Warszawa	mazowieckie	89-608	Płesno	pomorskie
00-002	Warszawa	mazowieckie	72-300	Baszewice	zachodniopomorskie
00-002	Warszawa	mazowieckie	29-145	Krzepin	świętokrzyskie
Kontynuacja na następnej stronie					

00-002	Warszawa	mazowieckie	20-244	Lublin	lubelskie
00-002	Warszawa	mazowieckie	32-540	Czyżówka	małopolskie
00-002	Warszawa	mazowieckie	67-312	Niegosławice	lubuskie
00-002	Warszawa	mazowieckie	16-506	Zelwa	podlaskie
00-002	Warszawa	mazowieckie	11-710	Nowy Żydrój	warmińsko-mazurskie
00-002	Warszawa	mazowieckie	37-310	Nowa Sarzyna	podkarpackie
00-002	Warszawa	mazowieckie	48-315	Mańkowice	opolskie
00-002	Warszawa	mazowieckie	62-563	Dąbrówka	wielkopolskie
00-002	Warszawa	mazowieckie	42-263	Huta Stara A	śląskie

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

W wyniku otrzymaliśmy niemal milion tras, które będą stanowiły podstawy do dalszych badań. Wszystkie trasy zostały oszacowane według parametrów samochodów ciężarowych. Dzięki temu uwzględniane są ograniczenia przejazdu dla pojazdów ciężarowych oraz koszty wynikające z poboru opłat na odcinkach płatnych. Oprócz podstawowych danych dotyczących adresów i ich lokalizacji, w wyniku obliczania tras otrzymaliśmy wiele innych zmiennych, które mogą zostać wykorzystane do dalszych analiz.

1.6.1 Obliczenie tras

Do przeliczenia tras dla samochodów ciężarowych został wykorzystany moduł planowania tras dostarczany przez PTV xServer. API PTV Route Planning uwzględnia przy obliczaniu optymalnej trasy wszystkie istotne atrybuty specyficzne dla samochodów ciężarowych, jak długość, wysokość, ciężar, ładunek specjalny i ograniczenia dostępu. Dodatkowo uwzględniane są bieżące informacje o ruchu drogowym i schematy ruchu, aby uniknąć zatorów i ominąć wąskie gardła. PTV umożliwia obliczanie dokładnych kosztów opłat drogowych w oparciu o atrybuty pojazdu podczas planowania tras i określanie, kiedy trasy wolne od opłat drogowych stanowią realną alternatywę. Jesteśmy w stanie również otrzymać dokładną emisję na podstawie atrybutów pojazdu i uznanych norm, takich jak CEN, HBEFA, CO₂e, COPERT.

Rozdział 2

Definicja i zastosowanie modelu grawitacyjnego

Dużą rolę przy tworzeniu modeli grawitacyjnych w ekonomicznej analizie przestrzennej, odgrywa prawo powszechnego ciążenia (gravitacji) Newtona, które powszechnie było wykorzystywane w fizyce. Izaak Newton (1642–1727) to angielski fizyk, astronom, matematyk, filozof i alchemik. Uznawany za jednego z najwybitniejszych i najważniejszych uczonych wszech czasów (Lima, 2017). Odkrywcą między innymi trzech zasad dynamiki, które brzmią następująco (Mendie, 2016):

1. W inercjalnym układzie odniesienia, jeśli na ciało nie działa żadna siła lub siły działające równoważą się, to ciało pozostaje w spoczynku lub porusza się ruchem jednostajnym prostoliniowym.
2. W inercjalnym układzie odniesienia jeśli siły działające na ciało nie równoważą się (czyli wypadkowa siła \vec{F}_w jest różna od zera), to ciało porusza się z przyspieszeniem wprost proporcjonalnym do siły wypadkowej, a odwrotnie proporcjonalnym do masy ciała.
3. Oddziaływania ciał są zawsze wzajemne. W inercjalnym układzie odniesienia siły wzajemnego oddziaływania dwóch ciał mają takie same wartości, taki sam kierunek, przeciwne zwroty i różne punkty przyłożenia (każda działa na inne ciało).

Innym ważnym odkryciem dokonany przez Newtona było prawa powszechnego ciążenia, które głosi: *„Każde dwa ciała przyciągają się wzajemnie. Siła wzajemnego przyciągania dwóch ciał jest wprost proporcjonalna do iloczynu ich mas i odwrotnie proporcjonalna*

do kwadratu odległości między ich środkami.” Oba te prawa przyczyniły się do powstania modelu grawitacyjnego, który jest narzędziem często wykorzystywanym w przestrzennych analizach danych. Model ten ma za cel wyjaśnienie wartości zmiennych za pomocą „mas” oraz „dystansu” pomiędzy nimi, co stanowiło analogię do równania grawitacji Newtona (Klimczak, 2015). Początkowo model ten okazał się bardzo dobrym narzędziem w wyjaśnianiu determinant handlu zagranicznego, natomiast znalazł on również swoje zastosowania w wielu innych branżach, dzięki czemu jego postaci ewoluowały.

2.1 Definicja modelu grawitacyjnego

Według autorów artykułu *„Analiza migracji wewnętrznych w Polsce z wykorzystaniem modelu grawitacji”* (Pietrzak i in., 2013), *„Modele grawitacyjne opisują zależność pomiędzy wielkością przepływu wybranej kategorii ekonomicznej a zmiennymi objaśniającymi”*. Modele te są wykorzystywane w badaniach z analizy przepływów, czy też wzajemnych oddziaływań pomiędzy poszczególnymi parami jednostek przestrzennych. W książce (Sucheckie, 2010) *„Ekonometria przestrzenna”* w modelach grawitacji badaniu poddane zostają wzajemne oddziaływania między poszczególnymi parami jednostek przestrzennych, dotyczące:

- migracja ludności,
- przepływy pieniężne,
- przepływy towarowe,
- przejazdy osobowe,
- ruch turystyczny,
- przestrzenna zmienność popytu, podaży i dochodu,
- efekty przepływów handlu międzynarodowego,
- obszarów dominacji miast.

Podstawą do utworzenia modelu grawitacyjnego jest ustalenie:

- jednostek przestrzennych, którymi mogą być: województwa, regiony, kraje,

- masy tych jednostek, czyli liczbę ludności, wielkość produkcji, PKB, liczbę zachowań, zatrudnienie, wielkość obrotu w handlu zagranicznym, wielkość inwestycji, sprzedaż detaliczną,
- wag jednostek przestrzennych, które można przyjmować jako z góry zadane lub wyznaczać na podstawie danych statystycznych; parametry statystyczne wag szacuje się tak jak pozostałe parametry strukturalne modelu, np. metodą najmniejszych kwadratów,
- odległości między jednostkami przestrzennymi (między ich środkami ciężkości), czyli: odległość fizyczną czy też czasową.

Za miarę dystansu można przyjąć: odległość geograficzną, zużycie energii, koszty podróży, czy też czas podróży. Natomiast wagami może być PKB na mieszkańca czy liczba mieszkańców (Popowych, 2018). Model grawitacji w postaci zależności funkcyjnej można przedstawić w następujący sposób (Suchecki, 2010):

$$Y_{ij} = \alpha_0 M_i^{\alpha_1} P_i^{\alpha_2} M_j^{\alpha_3} P_j^{\alpha_4} D_{ij}^{\alpha_5} e^{\varepsilon} \quad (2.1)$$

gdzie:

- Y_{ij} - wielkość oddziaływania między i-tą oraz j-ą jednostką przestrzenną, przepływ np. dóbr, migracje ludności,
- \mathbf{P} - wektor zmiennych charakteryzujący masy poszczególnych jednostek,
- \mathbf{M} - wektor wag tych mas,
- \mathbf{D} - macierz odległości między jednostkami,
- \mathbf{A} - wektor parametrów stałych i wykładników potęgowych mas i odległości,
- $\alpha_0, \dots, \alpha_5$ to parametry modelu, natomiast n jest liczbą jednostek przestrzennych,
- ε – składnik losowy modelu.

Model powszechnego ciążenia sprowadzono do postaci, w której masy zastąpiono lokalizacjami, mianowicie:

$$Y_{ij} = \alpha \beta_i P_i^{\alpha_k} \beta_j P_j^{\alpha_k} D_{ij}^{\alpha_k} \quad (2.2)$$

gdzie:

- **i** - lokalizacja i-ta,
- **j** - lokalizacja j-ta,
- α - stała modelu grawitacji
- β - parametry strukturalne mas i wag mas modelu.

Mimo intuicyjności i uniwersalności wzoru powszechnego ciążenia, istnieją również problemy związane z jego wykorzystywaniem. Jednym z nich jest dobranie zmiennych przedstawiających masy owego modelu oraz dystansu pomiędzy dwoma partnerami w wymianie towarowej (Heffner i Gibas, 2007). Masa była zazwyczaj przedstawiana w formie PKB lub populacji danego kraju/obszaru, a dystans natomiast poprzez fizyczną odległość pomiędzy nimi.

2.2 Grawitacyjna analiza handlu w Pythonie

Z uwagi na to, że w pracy do skonstruowania analizy emisyjności transportu drogowego wykorzystany został język Python, w niniejszym podrozdziale szerzej opisane zostaną funkcje programu, oraz pakiety pozwalające na tworzenie zaawansowanych analiz.

W Pythonie stworzona jest specjalna biblioteka, która znacznie ułatwia wykorzystanie modeli GME (Gravity Modeling Environment) do analizy handlu grawitacyjnego. Pakiet składa się z narzędzi, które pomagają w szybkiej, elastycznej i solidnej estymacji modeli grawitacyjnych przy użyciu nowoczesnych, najlepszych praktyk (Herman, 2021).

Ponieważ narzędzia są zaimplementowane w Pythonie, użytkownicy mają dostęp do ogromnej i wciąż rosnącej kolekcji narzędzi innych firm, które mogą włączyć do swojej pracy i rozszerzyć ją. Python jest zarówno elastycznym, potężnym i darmowym językiem programowania, który może być łatwo używany na szerokiej gamie komputerów bez ponoszenia kosztów.

Pakiet został stworzony przez Grupę Modelowania Grawitacyjnego przy Międzynarodowej Komisji Handlu Stanów Zjednoczonych oraz jest ogólnodostępny do pobrania z oficjalnej strony bibliotek języka Python: <https://pypi.org/project/gme/>. Pakiet GME wymaga również dodatkowych pakietów takich jak:

- pandas, który jest szybkim, potężnym, elastycznym i łatwym w użyciu narzędziem open source do analizy i manipulacji danymi, zbudowanym na bazie języka programowania Python.,
- statsmodels - jest modułem Pythona, który dostarcza klasy i funkcje do estymacji wielu różnych modeli statystycznych, jak również do przeprowadzania testów i eksploracji danych statystycznych. Dla każdego estymatora dostępna jest obszerna lista statystyk wynikowych. Wyniki są testowane w oparciu o istniejące pakiety statystyczne, aby zapewnić ich poprawność,
- patsy - jest pakietem Pythona służącym do opisywania modeli statystycznych (w szczególności modeli liniowych, lub modeli, które mają składnik liniowy) i budowania macierzy obliczeniowych,
- scipy, który jest opartym na Pythonie ekosystemem oprogramowania open-source dla matematyki, nauk ścisłych i inżynierii.

Jeśli nie są one jeszcze obecne w systemie, zostaną automatycznie zainstalowane podczas instalacji GME.

Postępowanie w ramach pakietu GME jest następujące:

- Wczytanie wymaganych do badania danych,
- Następnie należy utworzyć instancję obiektu EstimationData przy użyciu przykładowych danych. Aby utworzyć instancję EstimationData trzeba dostarczyć DataFrame i identyfikatory dla niektórych kluczowych kolumn, takich jak przepływy handlowe, importer/eksporter, rok i sektor (jeśli dotyczy),
- Po wytworzeniu obiektu EstimationData należy wytworzyć obiekt EstimationModel, który będzie używany do wytwarzania oszacowań grawitacyjnych. Aby utworzyć instancję EstimationModel, należy dostarczyć obiekt EstimationData oraz potrzebną specyfikację,

- Gdy EstimationModel jest już zdefiniowany, można go oszacować stosując metodę `.estimate()`. Wyniki są przechowywane w kolekcji (w Pythonie nazywanej słownikiem), gdzie każdy obiekt ma swój własny zestaw wyników.

W dalszych krokach należy wykorzystać dostępne w internecie biblioteki SHP, w celu naniesienia zwróconych przez model informacji. W tym badaniu, zostały wykorzystane pliki SHP dostępne pod adresem: <https://gis-support.pl/baza-wiedzy-2/dane-do-pobrania/granice-administracyjne/>. Do wyrysowania kartogramów wykorzystywana jest natomiast biblioteka `matplotlib.pyplot`, która w łatwy sposób umożliwia rysowanie grafów w języku Python. Dokładna metodologia i implementacja plików SHP, dostępna jest do wglądu w sekcji "Kody języka Python".

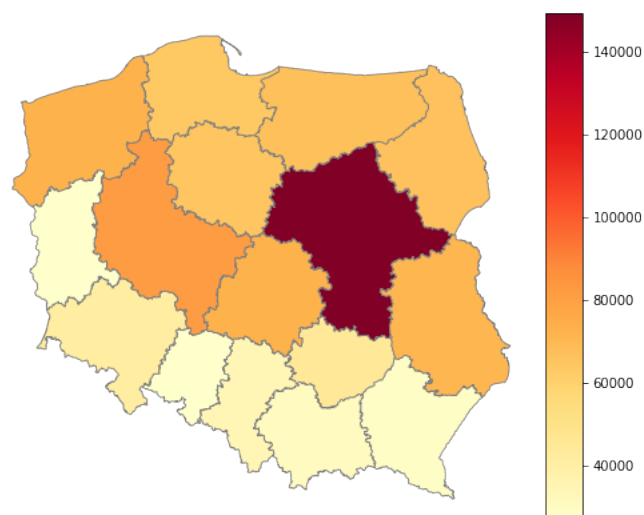
Rozdział 3

Estymacja modelu grawitacji w analizie emisji spalin samochodów ciężarowych

W niniejszym rozdziale przedstawiono wykorzystanie modelu grawitacji w analizie emisji spalin wydalaných przez samochody ciężarowe w logistyce. Na uzyskanie danych do modelu składało się wiele czynności związanych ze zbieraniem, przetwarzaniem i generowaniem danych geolokalizacyjnych dla wszystkich kodów pocztowych w Polsce. Następnie zostały wygenerowane przykładowe trasy, a na ich podstawie zostały utworzone poszczególne modele grawitacyjne dla konkretnych emisji pierwiastków chemicznych.

3.1 Analiza parametrów modelu grawitacji

Do badania zostało wykorzystane 943036 tras. Oczywiście ilość tras dla każdego województwa jest proporcjonalna do ilości kodów pocztowych w danym województwie, na których podstawie powstawały owe trasy. Największa liczba tras pod względem wyjazdu z danego województwa (miejsca załadunku) znajduje się w województwie mazowieckim (149424), wielkopolskim (81920) oraz łódzkie (72864). Województwa o najmniejszej liczbie tras wyjazdowych to województwo lubuskie (26320), opolskie (26480) oraz podkarpackie (28368). Ilość tras dla poszczególnych województw przedstawia rysunek 3.1.



Rysunek 3.1. Liczba tras wyjazdowych według województw w Polsce

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

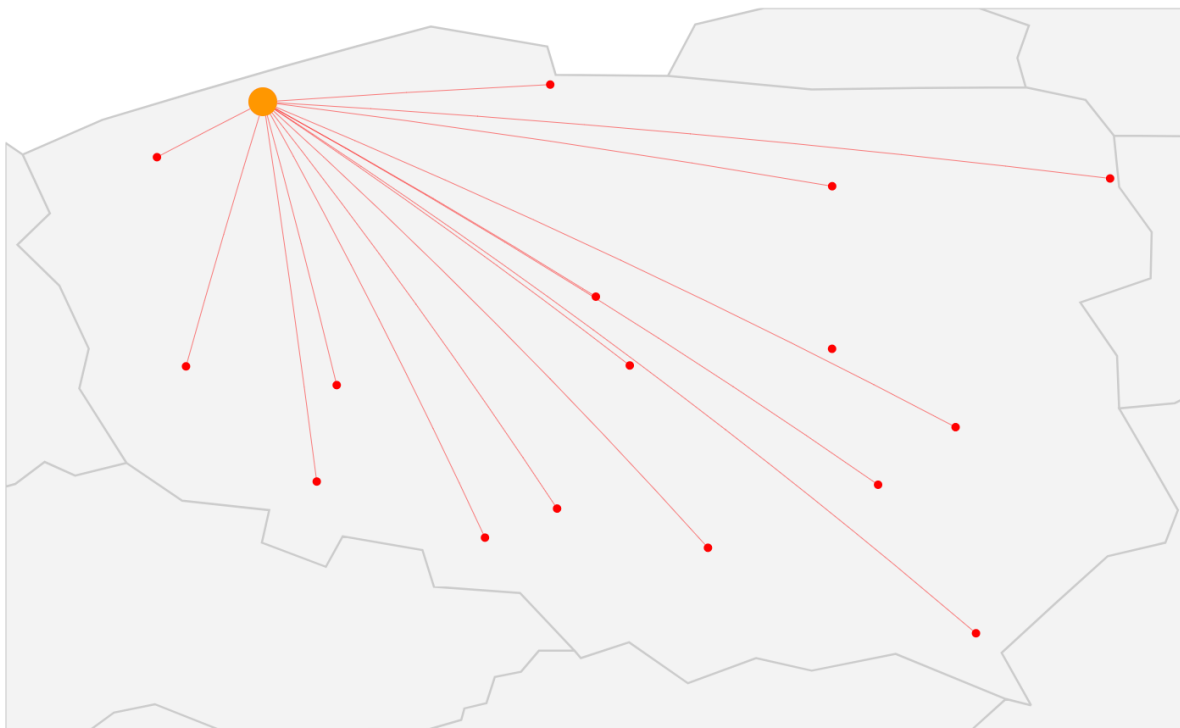
Dokładną ilość utworzonych tras pod względem wyjazdu z danego województwa zawiera poniższa tabela.

Tabela 3.1. Liczba utworzonych tras dla danego województwa

Województwo	Liczba tras
mazowieckie	149424
wielkopolskie	81920
łódzkie	72864
zachodniopomorskie	72736
lubelskie	70640
warmińsko-mazurskie	66784
podlaskie	66512
kujawsko-pomorskie	65056
pomorskie	64272
świętokrzyskie	45008
dolnośląskie	41264
śląskie	34576
małopolskie	30880
podkarpackie	28368
opolskie	26480
lubuskie	26320

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Dla każdego kodu pocztowego zostało losowo wygenerowanych 16 tras z innych województw.



Rysunek 3.2. Trasy dla 75-360, Koszalin

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Powyższy przykład przedstawia trasy wygenerowane dla kodu pocztowego 75-360 znajdującego się w Koszalinie, województwie zachodniopomorskim. Zostało mu przyporządkowanych 15 kodów pocztowych z innych województw oraz jeden kod pocztowy również z województwa zachodniopomorskiego.

Wszystkie trasy zostały obliczone za pomocą modułu planowania tras dostarczanego przez PTV¹. Głównymi zmiennymi dla obliczonych tras są dystans, czas, dystans odcinków płatnych oraz koszty odcinków płatnych. Podstawowe statystyki opisowe dla zmiennych przedstawia tabela 3.2.

¹<https://www.ptvgroup.com/en/solutions/products/ptv-xserver/developer-zone/route-planning-api/>

Tabela 3.2. Statystyki opisowe wygenerowanych tras

	Odległość (m)	Czas (s)	Odcinki płatne (m)	Koszt (€¢)	Manewry
mean	398667.72	28925.34	212345.57	1809.04	26.84
std	186204.12	17546.60	159744.62	1706.77	9.69
min	129.00	423.00	0.00	0.00	0.00
25%	258024.75	19516.00	80600.00	505.00	20.00
50%	390263.50	26028.00	187010.00	1286.00	26.00
75%	528352.25	32737.00	318702.50	2686.00	33.00
max	1146873.00	107053.00	787200.00	7401.00	108.00

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Średnia długość trasy wyniosła 398 667,72 m. Odchylenie standardowe zmiennej opisującej dystans, wyniosło 186 204,12 m, co wskazuje na dosyć szerokie rozrzucenie wokół jej średniej. Najkrótsza trasa ma 129 m, a najdłuższa 1 146 873 m. 75% tras ma długość mniejszą niż 528 352,25 m, 50% ma długość mniejszą niż 390 263,50, a mniej niż 258 024,75 m posiada 25% tras.

Czas pokonania danej trasy wyrażany jest w sekundach. Trasa średnio jest pokonywana w 8 h 2 min 5 s. Standardowe odchylenie od średniego czasu pokonania trasy wynosi 4 h 52 min 26 s. Najdłużej trwająca podróż samochodem ciężarowym wyniosła 29 h 44 min 13 s. 75% tras ma długość mniejszą niż 9 h 5 min 37 s, 50% mniejszą niż 7 h 13 min 48 s, a mniej niż 5 h 25 min 16 s posiada 25% tras.

Jako że obliczenia są z uwzględnieniem samochodów ciężarowych, to zostały również obliczone odległości odcinków płatnych oraz ich koszty. Średnia długość odcinków na trasach to 212 345,57 m. Występowały również sytuacje, gdzie nie były wykorzystywane żadne odcinki płatne i w tym przypadku ich długość wyniosła 0. Najdłuższe odcinki, za które kierowca powinien uiścić opłatę to 787 200 m.

Z faktu iż narzędzie, które zostało wykorzystane do obliczeń pokonania tras, działa na arenie międzynarodowej (głównie w europie), to koszty pokonania trasy zostały zaprezentowane w Euro centach. Są to koszty wynikające wyłącznie z opłat drogowych i nie są one w żaden sposób związane z zakupem paliwa na potrzeby podróży. Średni koszt pokonania odcinków płatnych w naszym zbiorze to 1809 Euro Centów, co w przeliczeniu na PLN przy kursie 4.58 (na dzień 24/08/2021) wynosi 82,77 zł. Tak samo jak

z płatnymi odcinkami, minimalny koszt wynosi 0 zł. Jest to sytuacja, gdzie kierowca nie korzystał na swojej drodze z żadnych odcinków płatnych dla samochodów ciężarowych. Maksymalny koszt jaki winniśmy ponieść to 338,64 zł.

Ostatnią zmienną, której się przyjrzymy są manewry wykonywane podczas przejazdu. Są to wszelkiego rodzaju zmiany kierunku jazdy, skrzyżowania o ruchu okrężnym, czy zmiany pierwszeństwa, które wymagają zwiększonej aktywności i ostrożności. Średnia ilość manewrów wykonywanych przez kierowce podczas podróży to w zaokrągleniu 27. Krótkie trasy na prostych odcinkach, nie wymagających wielu akcji powodują, że minimalna ilość manewrów wyniosła 0. Maksymalnie musiałyby zostać przeprowadzonych 108 manewrów na trasie. Odchylenie standardowe dla tej zmiennej wyniosło 9,69.

3.2 Analiza porównawcza parametrów modelu ciążenia

Do budowy modeli grawitacyjnych dla poszczególnych emisji, służyć nam będą poniższe zmienne. Są one wynikiem kalkulacji tras przez serwis PTV.

- **Dystans** - zmienna opisująca odległość wyrażoną w metrach, konieczną do pokonania na danej trasie,
- **Czas** - czas potrzebny do pokonania danej trasy wyrażony w sekundach,
- **Długość odcinków płatnych** - długość odcinków płatnych na trasie, wyrażona w metrach,
- **Suma kosztów odcinków płatnych** - koszty odcinków płatnych na trasie, według stawek dla pojazdów ciężarowych, wyrażona w euro centach,
- **Manewry** - liczba manewrów wykonanych na danej trasie.

Natomiast zmienne, dla których będą tworzone modele to:

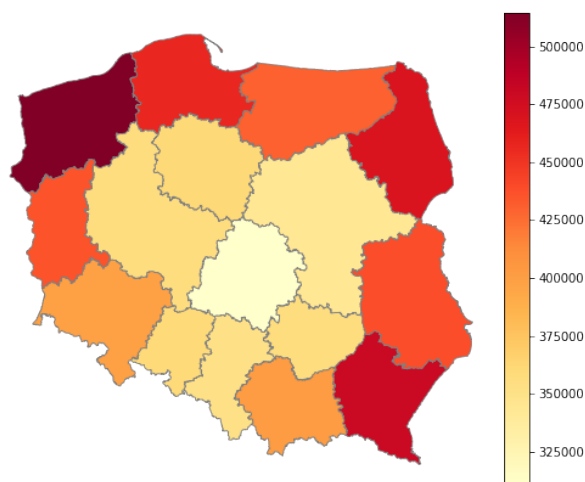
- **Spalanie paliwa (Diesel),**
- **Węglowodory,**
- **Metan,**
- **Węglowodory z wyłączeniem metanu,**

- Tlenek węgla,
- Dwutlenek węgla,
- Dwutlenek siarki,
- Tlenek azotu,
- Podtlenek azotu,
- Amoniak,
- Benzen,
- Cząstki stałe.

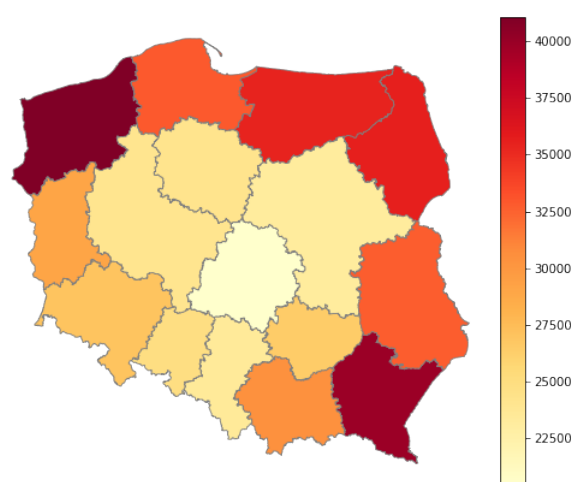
Tabela 3.3. Średnie wartości parametrów modelu z podziałem na województwa

	distance_m	time_s	toll_distance_m	toll_cost_euro	manoeuvres
dolnośląskie	398,835.7258	27,035.8476	239,284.9285	1,837.2401	26.3134
kujawsko-pomorskie	360,753.3658	24,191.9536	189,790.9784	1,465.4528	24.7982
lubelskie	438,205.3791	32,715.4105	214,107.9307	1,812.2670	32.8071
lubuskie	436,089.6925	29,120.0353	284,916.6956	3,108.4330	24.3472
mazowieckie	343,771.3784	23,236.0283	188,673.7382	1,558.8460	25.8296
małopolskie	402,122.1614	30,505.7869	212,576.8208	2,003.0287	27.0200
opolskie	359,815.5621	24,926.8027	183,796.1179	1,443.4877	27.7648
podkarpackie	480,006.5964	39,872.8015	266,535.6371	2,378.4743	29.9319
podlaskie	469,553.7622	35,660.4736	237,561.2848	1,851.5700	26.2572
pomorskie	456,607.6921	32,901.8575	267,194.0573	2,026.8059	26.4349
warmińsko-mazurskie	430,253.3522	35,437.3877	157,766.1858	1,128.8054	31.6277
wielkopolskie	356,050.1489	24,129.6744	192,905.5712	1,919.0719	25.9788
zachodniopomorskie	514,402.9772	41,058.4575	293,199.1216	2,711.6434	25.6596
łódzkie	310,140.1422	20,444.7765	178,543.7919	1,602.0477	24.2497
śląskie	351,221.5358	23,491.5180	185,411.3887	1,486.3421	24.5812
świętokrzyskie	356,767.0086	26,420.7643	181,634.8033	1,628.2119	26.8656

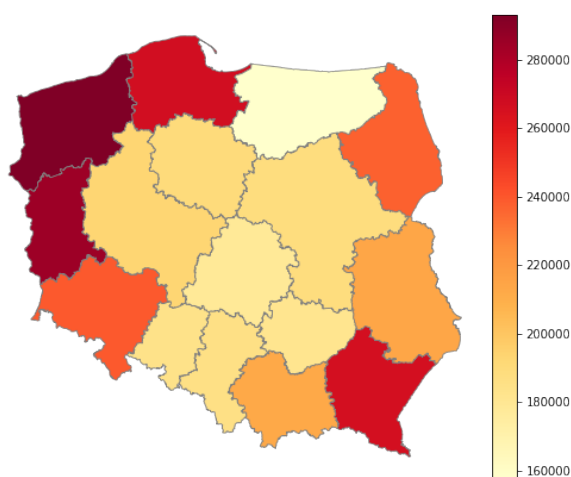
Źródło: Opracowanie własne w języku Python na podstawie zebranych danych



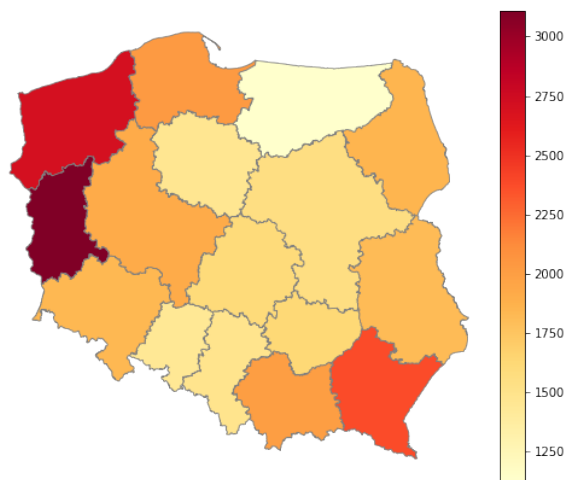
(a) Długość trasy



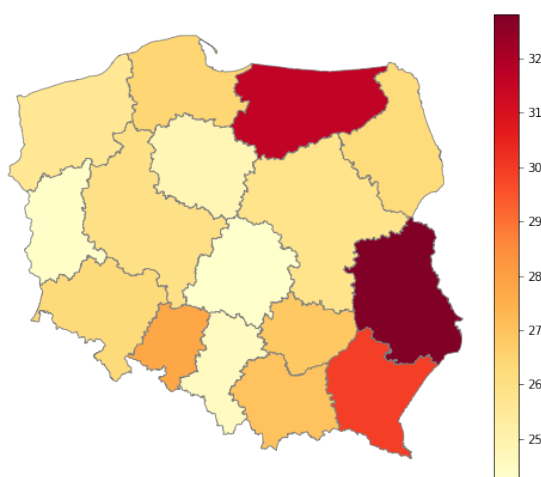
(b) Czas



(c) Długość odcinków płatnych



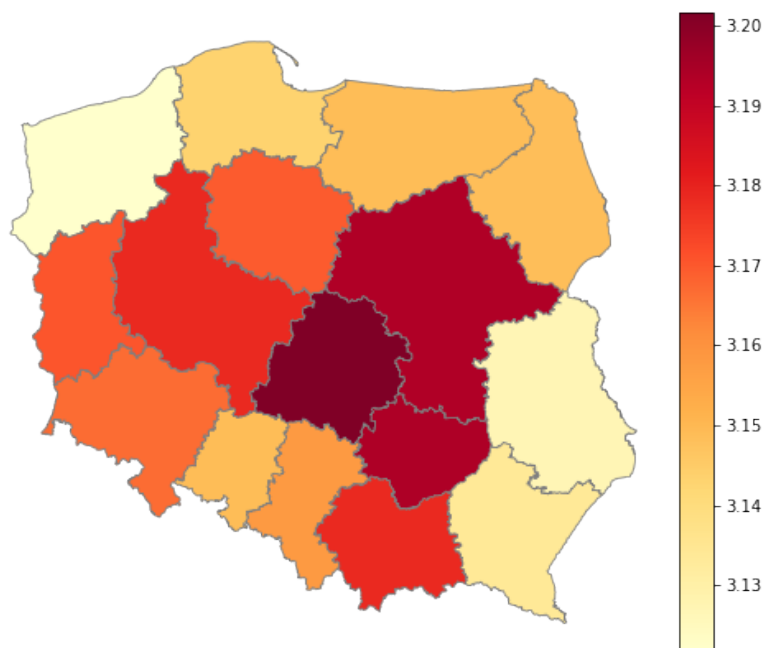
(d) Koszt odcinków płatnych



(e) Manewry na trasie

Rysunek 3.3. Średnie wartości poszczególnych parametrów modelu w podziale na województwa

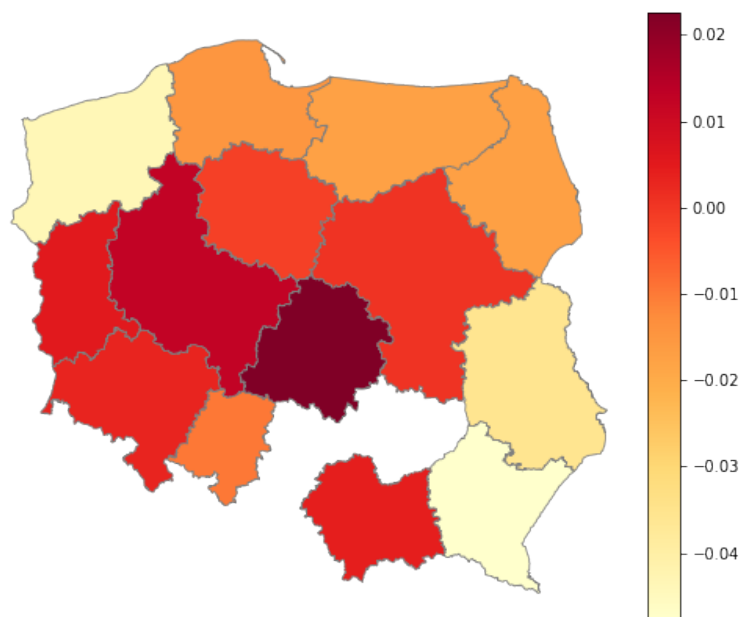
3.2.1 Zużycie paliwa



Rysunek 3.4. Współczynnik spalania paliwa według województw wyjazdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Zużycie paliwa często stanowi jedno z podstawowych kryteriów wyboru pojazdu. Pomimo najbardziej oczywistego kosztu, związanego ze spalaniem - cena paliwa, również uzależnione są opłaty od poruszania się pojazdów po niektórych drogach. Dotyczy to przede wszystkim pojazdów ciężarowych i autobusów. Na spalanie duży wpływ ma również pokonywana trasa. Występujące na niej częste zakręty, zmuszające do przyhamowań i następnie przyśpieszeń, podjazdy pod wzniesienia, zmiany wysokości terenu, czy też jakość i zaawansowanie infrastruktury drogowej. Pierwszy z utworzonych modeli przedstawia, w jaki sposób zużycie paliwa przy transporcie krajowym, jest zależne od województwa. Rysunek 3.4 prezentuje kartogram współczynnika spalania paliwa dla tras, które swój początek miały w danym województwie (województwie początku trasy - wyjazdowym). Na pierwszy rzut oka widać, że najwyższe współczynniki spalania występują w centralnej Polsce. Duży wpływ może mieć tutaj średnia długość trasy dla tych województw. Im krótsza trasa, tym mniejsze rozłożenie spalania na dłuższych, prostych odcinkach, gdzie zużycie paliwa powinno być względnie mniejsze. Najmniejszy poziom zużycia paliwa występuje w zachodniopomorskim, co spowodowane może być najwyższymi średnimi wartościami długości trasy oraz wykorzystanych odcinków płatnych. Dzięki użytkowaniu płatnych autostrad i ruch szybkiego ruchu, kierowca nie



Rysunek 3.5. Współczynnik spalania paliwa według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

musi wykonywać wielu manewrów, dzięki temu spalanie pojazdu powinno utrzymywać się na stałym, niskim poziomie. Kartogram 3.5 prezentuje te same współczynniki, tylko w ujęciu o województwa przyjezdne. Widać, że rozkład spalania paliwa kształtuje się w bardzo podobny sposób co do województw wyjezdnych. W drugim kartogramie widzimy, że uwzględnione nie zostały dwa województwa - świętokrzyskie oraz śląskie. Wynika to z wykluczenia tych dwóch województw przez model podczas estymacji. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów. Model wyklucza z estymacji województwa, gdzie przepływ pomiędzy danymi kontrahentami (importerami oraz eksporterami) jest zerowy. Województwo śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. *„Zmienne idealnie współliniowe to zmienne przewidujące w tym samym modelu regresji, które są skorelowane i nie mogą one niezależnie przewidywać wartości zmiennej zależnej. Innymi słowy, wyjaśniają one część tej samej wariancji w zmiennej zależnej, co z kolei zmniejsza ich istotność statystyczną”*(Enders, 2013). Dzięki temu model regresji wielorakiej ze współliniowymi predyktorami może wskazywać, jak dobrze cała wiązka predyktorów przewiduje zmienną wynikową, ale może nie dawać ważnych wyników na temat każdego indywidualnego predyktora lub na temat tego, które predyktory są zbędne w stosunku do innych².

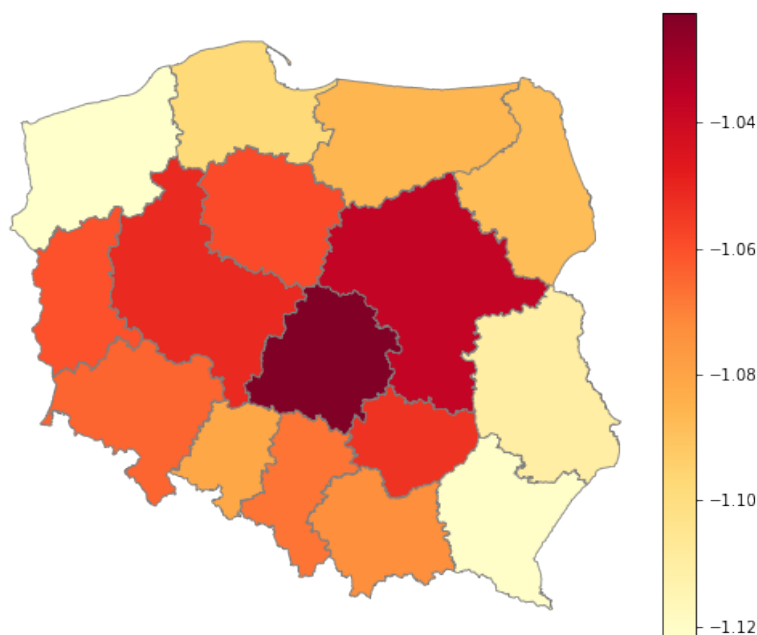
²<https://en.wikipedia.org/wiki/Multicollinearity>

Tabela 3.4. Oszacowania współczynników modelu dla zużycia paliwa

	coef	std err	t	P> t	[0.025	0.975]
from_dolnośląskie	3.1667	0.001	3180.888	0.000	3.165	3.169
from_kujawsko-pomorskie	3.1695	0.001	3564.415	0.000	3.168	3.171
from_lubelskie	3.1270	0.001	3113.398	0.000	3.125	3.129
from_lubuskie	3.1706	0.001	3054.270	0.000	3.169	3.173
from_mazowieckie	3.1938	0.001	3970.985	0.000	3.192	3.195
from_małopolskie	3.1788	0.001	3013.908	0.000	3.177	3.181
from_opolskie	3.1489	0.001	2676.899	0.000	3.147	3.151
from_podkarpackie	3.1340	0.001	2499.109	0.000	3.132	3.136
from_podlaskie	3.1486	0.001	3154.433	0.000	3.147	3.151
from_pomorskie	3.1432	0.001	3195.560	0.000	3.141	3.145
from_warminsko-mazurskie	3.1489	0.001	3050.430	0.000	3.147	3.151
from_wielkopolskie	3.1786	0.001	3716.480	0.000	3.177	3.180
from_zachodniopomorskie	3.1216	0.001	3183.362	0.000	3.120	3.124
from_łódzkie	3.2017	0.001	3865.075	0.000	3.200	3.203
from_śląskie	3.1581	0.001	2911.890	0.000	3.156	3.160
from_świętokrzyskie	3.1941	0.001	3503.025	0.000	3.192	3.196
to_dolnośląskie	0.0031	0.001	5.507	0.000	0.002	0.004
to_kujawsko-pomorskie	-0.0015	0.001	-2.586	0.010	-0.003	-0.000
to_lubelskie	-0.0356	0.001	-61.720	0.000	-0.037	-0.034
to_lubuskie	0.0055	0.001	8.854	0.000	0.004	0.007
to_mazowieckie	0.0007	0.001	1.080	0.280	-0.001	0.002
to_małopolskie	0.0042	0.001	7.568	0.000	0.003	0.005
to_opolskie	-0.0096	0.001	-16.920	0.000	-0.011	-0.009
to_podkarpackie	-0.0479	0.001	-68.088	0.000	-0.049	-0.046
to_podlaskie	-0.0172	0.001	-30.491	0.000	-0.018	-0.016
to_pomorskie	-0.0146	0.001	-25.127	0.000	-0.016	-0.013
to_warminsko-mazurskie	-0.0174	0.001	-30.296	0.000	-0.018	-0.016
to_wielkopolskie	0.0125	0.001	21.569	0.000	0.011	0.014
to_zachodniopomorskie	-0.0437	0.001	-65.799	0.000	-0.045	-0.042
to_łódzkie	0.0226	0.001	34.131	0.000	0.021	0.024

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.2 Amoniak



Rysunek 3.6. Współczynnik emisji amoniaku według województw wyjezdnych

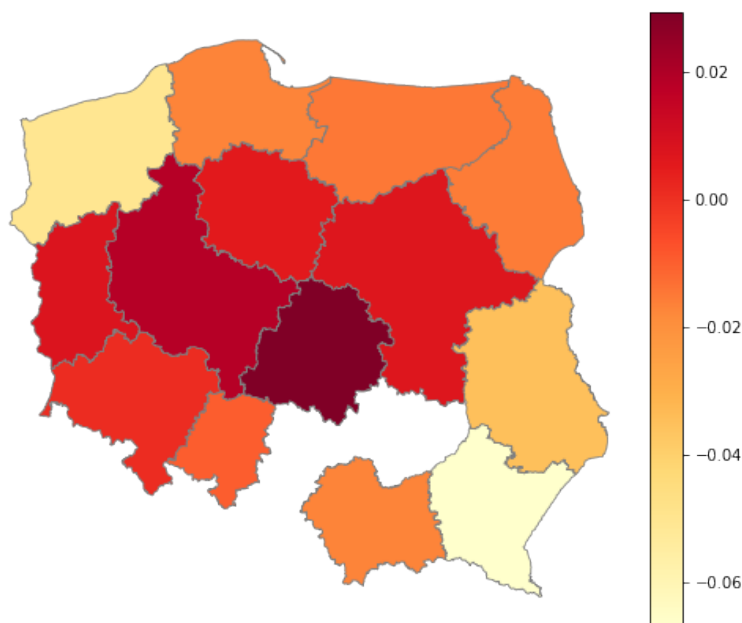
Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Amoniak znajdujący się w powietrzu atmosferycznym reaguje z unoszącymi się w powietrzu cząsteczkami takimi jak kwas azotowy tworząc unoszące się w powietrzu cząsteczki wielkości drobin kurzu. Cząsteczki te mogą zaostrzać stany chorobowe przy chorobach płuc i serca, a ponieważ są one higroskopijne, mogą również tworzyć podobną do smogu mgłę. Emisja amoniaku przez pojazdy, to obecnie jedynie niewielki procent całej emisji amoniaku, której większość pochodzi z fabryk oraz jest wydzielana przez zwierzęta hodowlane (Sagan, 2007). Amoniak przyczynia się do tworzenia aerozoli atmosferycznych, które mogą prowadzić do ograniczenia widzialności³. Rysunek 3.6 prezentuje oszacowane współczynniki emisji amoniaku dla województw wyjezdnych. Rozkład emisji prezentuje się w podobny sposób jak zużycie paliwa. Najwyższy współczynnik osiągnęło województwo łódzkie, a najniższy zachodniopomorskie. Zależność emisji amoniaku może odnosić się do tych samych przyczyn co przy spalaniu paliwa, ale może również być z nim bezpośrednio związane.

Mylącym przy interpretacji, mogą się okazać ujemne wartości stojące przy poszczególnych współczynnikach modelu. Według oficjalnej dokumentacji wykorzystywanej do

³<https://powietrze.uni.wroc.pl/base/t/amoniak-NH3>

badan biblioteki w modelu wykorzystywana jest funkcja wykładnicza⁴, która każdej ujemnej wartości, przyporządkowuje nieujemny wynik. Dzięki temu ujemne wyniki modelu, możemy przedstawić w zbiorze liczb nieujemnych, co znacznie ułatwia ich dalszą interpretację statystyczną oraz ekonomiczną.



Rysunek 3.7. Współczynnik emisji amoniaku według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Podobnie jak przy zużyciu paliwa, nieuwzględnione zostały dwa województwa - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Warto zaznaczyć, że prawie wszystkie zmienne występujące w modelu są istotne. Wyjątkiem jest współczynnik województwa dolnośląskiego dla tras przyjezdnych. Wartość p dla tej zmiennej wyniosła ponad 5%.

Tabela 3.5. Oszacowania współczynników modelu dla emisji amoniaku

	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0034	3.24e-06	1062.564	0.000	0.003	0.003
time_h	-0.0245	5.57e-05	-439.930	0.000	-0.025	-0.024
toll_distance_km	-0.0004	2.97e-06	-122.366	0.000	-0.000	-0.000
toll_cost_euro	-0.0010	1.8e-05	-56.260	0.000	-0.001	-0.001

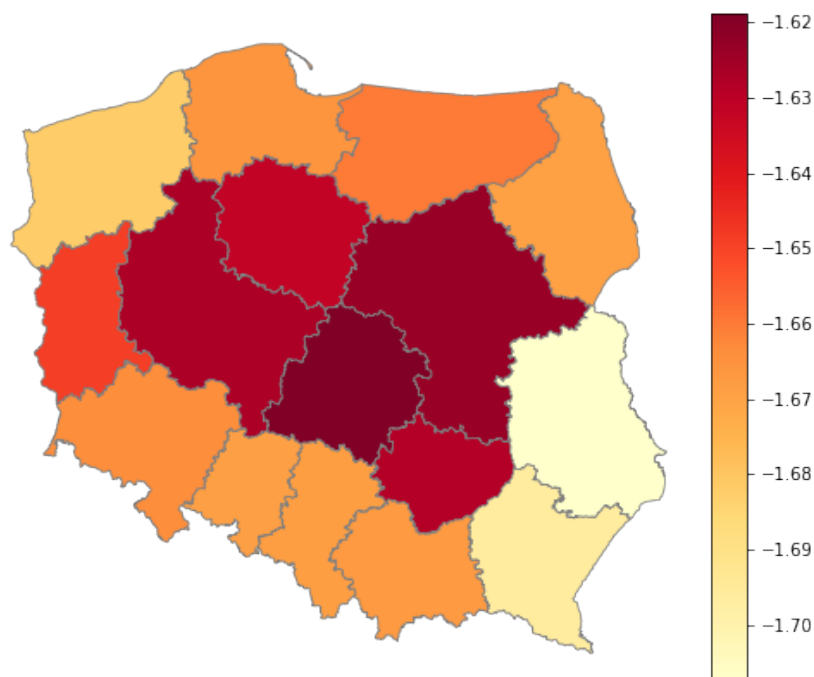
Kontynuacja na następnej stronie

⁴https://www.usitc.gov/data/gravity/gme_docs/estimate_technical/

manoeuvres	0.0025	1.77e-05	142.360	0.000	0.002	0.003
from_dolnośląskie	-1.0644	0.001	-1041.003	0.000	-1.066	-1.062
from_kujawsko-pomorskie	-1.0590	0.001	-1156.862	0.000	-1.061	-1.057
from_lubelskie	-1.1099	0.001	-1075.035	0.000	-1.112	-1.108
from_lubuskie	-1.0603	0.001	-989.879	0.000	-1.062	-1.058
from_mazowieckie	-1.0370	0.001	-1257.369	0.000	-1.039	-1.035
from_małopolskie	-1.0727	0.001	-981.999	0.000	-1.075	-1.071
from_opolskie	-1.0810	0.001	-894.830	0.000	-1.083	-1.079
from_podkarpackie	-1.1209	0.001	-861.180	0.000	-1.123	-1.118
from_podlaskie	-1.0879	0.001	-1056.793	0.000	-1.090	-1.086
from_pomorskie	-1.0975	0.001	-1083.237	0.000	-1.099	-1.096
from_warminsko-mazurskie	-1.0859	0.001	-1027.231	0.000	-1.088	-1.084
from_wielkopolskie	-1.0511	0.001	-1190.885	0.000	-1.053	-1.049
from_zachodniopomorskie	-1.1219	0.001	-1110.003	0.000	-1.124	-1.120
from_łódzkie	-1.0226	0.001	-1201.508	0.000	-1.024	-1.021
from_śląskie	-1.0674	0.001	-954.967	0.000	-1.070	-1.065
from_świętokrzyskie	-1.0533	0.001	-1117.936	0.000	-1.055	-1.051
to_dolnośląskie	0.0011	0.001	1.911	0.056	-2.79e-05	0.002
to_kujawsko-pomorskie	0.0055	0.001	9.164	0.000	0.004	0.007
to_lubelskie	-0.0350	0.001	-58.535	0.000	-0.036	-0.034
to_lubuskie	0.0078	0.001	12.373	0.000	0.007	0.009
to_mazowieckie	0.0070	0.001	10.894	0.000	0.006	0.008
to_małopolskie	-0.0168	0.001	-29.804	0.000	-0.018	-0.016
to_opolskie	-0.0094	0.001	-16.491	0.000	-0.011	-0.008
to_podkarpackie	-0.0666	0.001	-91.501	0.000	-0.068	-0.065
to_podlaskie	-0.0156	0.001	-26.712	0.000	-0.017	-0.014
to_pomorskie	-0.0171	0.001	-28.731	0.000	-0.018	-0.016
to_warminsko-mazurskie	-0.0148	0.001	-25.339	0.000	-0.016	-0.014
to_wielkopolskie	0.0188	0.001	31.340	0.000	0.018	0.020
to_zachodniopomorskie	-0.0502	0.001	-73.528	0.000	-0.052	-0.049
to_łódzkie	0.0294	0.001	43.313	0.000	0.028	0.031

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.3 Benzen

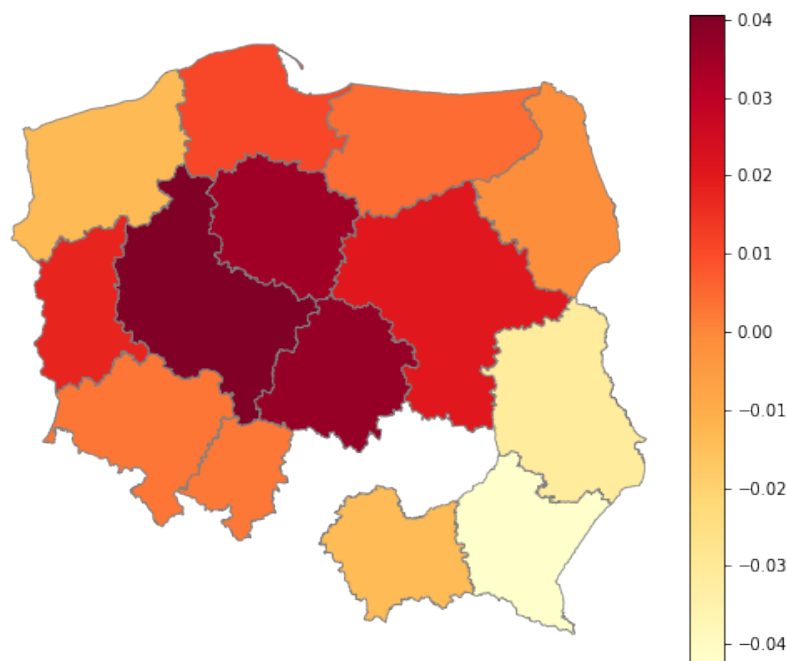


Rysunek 3.8. Współczynnik emisji benzenu według województw wyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Następnym związkiem chemicznym, który został poddany badaniu jest benzen. Benzen jest naturalnym składnikiem ropy naftowej. Stanowi jeden z podstawowych produktów przemysłu petrochemicznego. Długotrwałe narażenie na wysokie stężenie benzenu w powietrzu jest odpowiedzialne, między innymi, za potencjalnie śmiertelną białaczkę⁵. Pod względem emisji benzenu do atmosfery, również przoduje województwo łódzkie, dla którego jest najwyższy wskaźnik emisji benzenu (-1.6188). Jednak województwem o najmniejszej wartości jest województwo lubelskie. Ciekawym zjawiskiem jest to, że województwo lubelskie ma największą średnią ilość manewrów wykonywanych podczas pokonywania tras, które w teorii powinno się przełożyć na najwyższe emitowanie spalin.

⁵<https://www.poradnikzdrowie.pl/zdrowie/urazy-wypadki/benzen-toksycznosc-i-objawy-zatrucia-aa-SbFP-8ZX2-nXHg.html>



Rysunek 3.9. Współczynnik emisji benzenu według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Podobnie jak przy zużyciu paliwa, nieuwzględnione zostały dwa województwa - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Wszystkie współczynniki występujące w modelu są statystycznie istotne. Największą wartość p, osiągnął parametr "to_podlaskie" - 4%. Błąd standardowy dla wszystkich parametrów ukształtował się na poziomie 0.001.

Tabela 3.6. Oszacowania współczynników modelu dla emisji benzenu

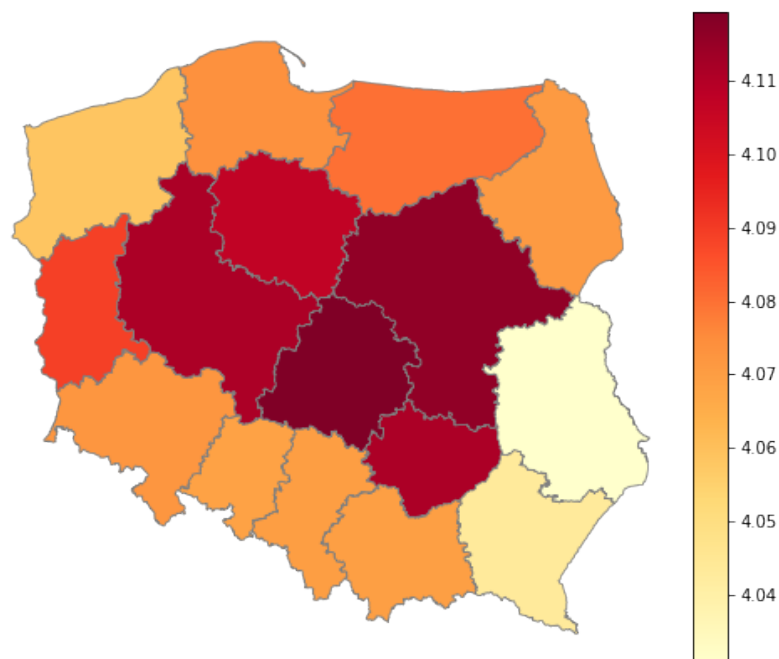
	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0031	3.23e-06	954.867	0.000	0.003	0.003
time_h	-0.0176	5.43e-05	-323.162	0.000	-0.018	-0.017
toll_distance_km	-0.0005	3.03e-06	-178.127	0.000	-0.001	-0.001
toll_cost_euro	-0.0004	1.68e-05	-23.184	0.000	-0.000	-0.000
manoeuvres	0.0045	1.8e-05	246.927	0.000	0.004	0.004
from_dolnośląskie	-1.6635	0.001	-1670.450	0.000	-1.665	-1.662
from_kujawsko-pomorskie	-1.6316	0.001	-1834.044	0.000	-1.633	-1.630
from_lubelskie	-1.7078	0.001	-1704.846	0.000	-1.710	-1.706
from_lubuskie	-1.6488	0.001	-1602.633	0.000	-1.651	-1.647

Kontynuacja na następnej stronie

from_mazowieckie	-1.6236	0.001	-2027.682	0.000	-1.625	-1.622
from_małopolskie	-1.6674	0.001	-1594.104	0.000	-1.669	-1.665
from_opolskie	-1.6688	0.001	-1440.796	0.000	-1.671	-1.667
from_podkarpackie	-1.6962	0.001	-1379.896	0.000	-1.699	-1.694
from_podlaskie	-1.6691	0.001	-1696.530	0.000	-1.671	-1.667
from_pomorskie	-1.6655	0.001	-1708.855	0.000	-1.667	-1.664
from_warminsko-mazurskie	-1.6601	0.001	-1610.887	0.000	-1.662	-1.658
from_wielkopolskie	-1.6269	0.001	-1927.644	0.000	-1.629	-1.625
from_zachodniopomorskie	-1.6821	0.001	-1738.636	0.000	-1.684	-1.680
from_łódzkie	-1.6188	0.001	-1969.985	0.000	-1.620	-1.617
from_śląskie	-1.6682	0.001	-1546.740	0.000	-1.670	-1.666
from_świętokrzyskie	-1.6284	0.001	-1793.833	0.000	-1.630	-1.627
to_dolnośląskie	0.0026	0.001	4.539	0.000	0.001	0.004
to_kujawsko-pomorskie	0.0349	0.001	58.548	0.000	0.034	0.036
to_lubelskie	-0.0315	0.001	-53.346	0.000	-0.033	-0.030
to_lubuskie	0.0178	0.001	29.008	0.000	0.017	0.019
to_mazowieckie	0.0205	0.001	31.535	0.000	0.019	0.022
to_małopolskie	-0.0139	0.001	-25.636	0.000	-0.015	-0.013
to_opolskie	0.0023	0.001	4.055	0.000	0.001	0.003
to_podkarpackie	-0.0426	0.001	-62.039	0.000	-0.044	-0.041
to_podlaskie	-0.0012	0.001	-2.056	0.040	-0.002	-5.51e-05
to_pomorskie	0.0110	0.001	18.350	0.000	0.010	0.012
to_warminsko-mazurskie	0.0045	0.001	7.338	0.000	0.003	0.006
to_wielkopolskie	0.0406	0.001	69.612	0.000	0.039	0.042
to_zachodniopomorskie	-0.0136	0.001	-20.305	0.000	-0.015	-0.012
to_łódzkie	0.0365	0.001	55.111	0.000	0.035	0.038

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.4 Tlenek węgla

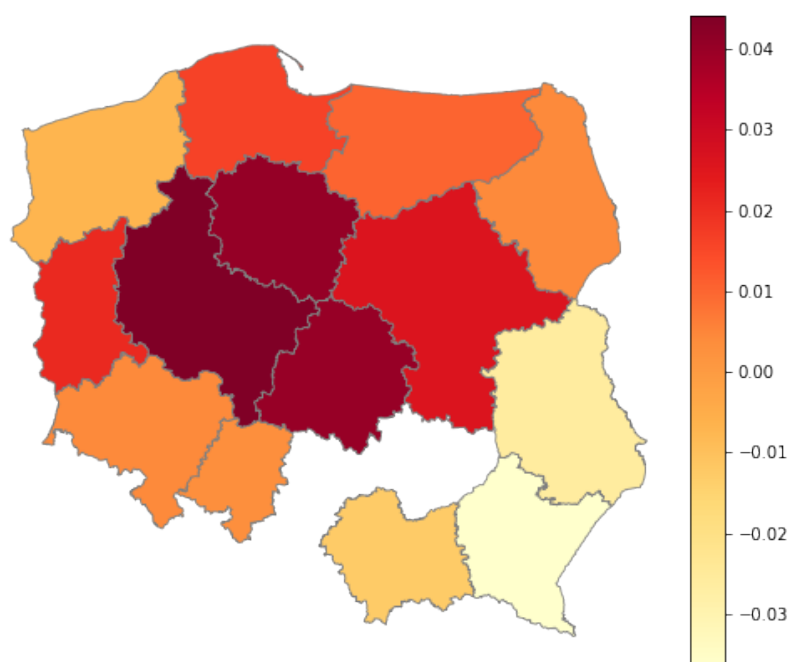


Rysunek 3.10. Współczynnik emisji tlenku węgla według województw wyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Tlenek węgla (CO), potocznie zwany czadem, jest bezwonny, bezbarwny, pozbawionym smaku silnie trującym gazem. Powstaje na skutek niepełnego spalania wszystkich paliw, m.in. gazu, węgla, drewna, ropy, oleju, benzyny, które spowodowane jest niedostateczną ilością tlenu niezbędnego do pełnego spalania paliw. Określany jest „cichym zabójcą”, ponieważ człowiek nie jest w stanie go wykryć za pomocą zmysłów⁶. Tlenek węgla niewiele się różni swoim rozkładem od poprzednich współczynników. Największą wartość względem województw wyjazdowych osiągnęło również województwo łódzkie (4.1192), lecz niedużo od niego różni się województwo mazowieckie (4.1159). Na końcu stawki ponownie jest województwo lubelskie (4.0308).

⁶<https://kadimex.pl/2017/09/08/tlenek-wegla-czym-jest-jak-powstaje-jak-mozna-sie-przed-nim-uchronic/>



Rysunek 3.11. Współczynnik emisji tlenku węgla według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Nieuwzględnione zostały dwa województwa przyjazdowe (końcowe) - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Wszystkie współczynniki występujące w modelu są statystycznie istotne, gdyż żaden z nich nie osiągnął wartości p większej niż 5%.

Tabela 3.7. Oszacowania współczynników modelu dla emisji tlenku węgla

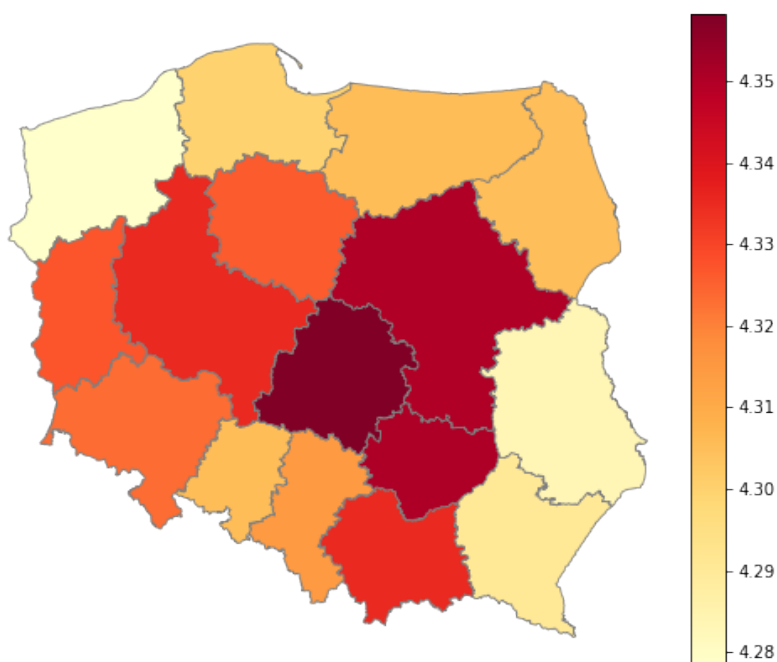
	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0031	3.32e-06	928.595	0.000	0.003	0.003
time_h	-0.0171	5.47e-05	-312.753	0.000	-0.017	-0.017
toll_distance_km	-0.0006	3.13e-06	-186.306	0.000	-0.001	-0.001
toll_cost_euro	-0.0004	1.69e-05	-21.415	0.000	-0.000	-0.000
manoeuvres	0.0046	1.85e-05	251.752	0.000	0.005	0.005
from_dolnośląskie	4.0724	0.001	4065.394	0.000	4.070	4.074
from_kujawsko-pomorskie	4.1071	0.001	4577.443	0.000	4.105	4.109
from_lubelskie	4.0308	0.001	3999.966	0.000	4.029	4.033
from_lubuskie	4.0888	0.001	3942.295	0.000	4.087	4.091
from_mazowieckie	4.1159	0.001	5105.442	0.000	4.114	4.117

Kontynuacja na następnej stronie

from_małopolskie	4.0697	0.001	3866.952	0.000	4.068	4.072
from_opolskie	4.0685	0.001	3499.325	0.000	4.066	4.071
from_podkarpackie	4.0435	0.001	3279.295	0.000	4.041	4.046
from_podlaskie	4.0710	0.001	4115.474	0.000	4.069	4.073
from_pomorskie	4.0738	0.001	4156.356	0.000	4.072	4.076
from_warminsko-mazurskie	4.0799	0.001	3926.533	0.000	4.078	4.082
from_wielkopolskie	4.1110	0.001	4837.118	0.000	4.109	4.113
from_zachodniopomorskie	4.0585	0.001	4162.616	0.000	4.057	4.060
from_łódzkie	4.1192	0.001	4982.430	0.000	4.118	4.121
from_śląskie	4.0700	0.001	3757.613	0.000	4.068	4.072
from_świętokrzyskie	4.1113	0.001	4494.407	0.000	4.109	4.113
to_dolnośląskie	0.0044	0.001	7.526	0.000	0.003	0.006
to_kujawsko-pomorskie	0.0406	0.001	67.069	0.000	0.039	0.042
to_lubelskie	-0.0260	0.001	-43.705	0.000	-0.027	-0.025
to_lubuskie	0.0212	0.001	34.002	0.000	0.020	0.022
to_mazowieckie	0.0261	0.001	39.632	0.000	0.025	0.027
to_małopolskie	-0.0126	0.001	-23.017	0.000	-0.014	-0.012
to_opolskie	0.0034	0.001	5.883	0.000	0.002	0.005
to_podkarpackie	-0.0365	0.001	-52.879	0.000	-0.038	-0.035
to_podlaskie	0.0044	0.001	7.682	0.000	0.003	0.006
to_pomorskie	0.0164	0.001	26.973	0.000	0.015	0.018
to_warminsko-mazurskie	0.0104	0.001	16.904	0.000	0.009	0.012
to_wielkopolskie	0.0442	0.001	74.847	0.000	0.043	0.045
to_zachodniopomorskie	-0.0071	0.001	-10.441	0.000	-0.008	-0.006
to_łódzkie	0.0404	0.001	60.476	0.000	0.039	0.042

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.5 Dwutlenek węgla

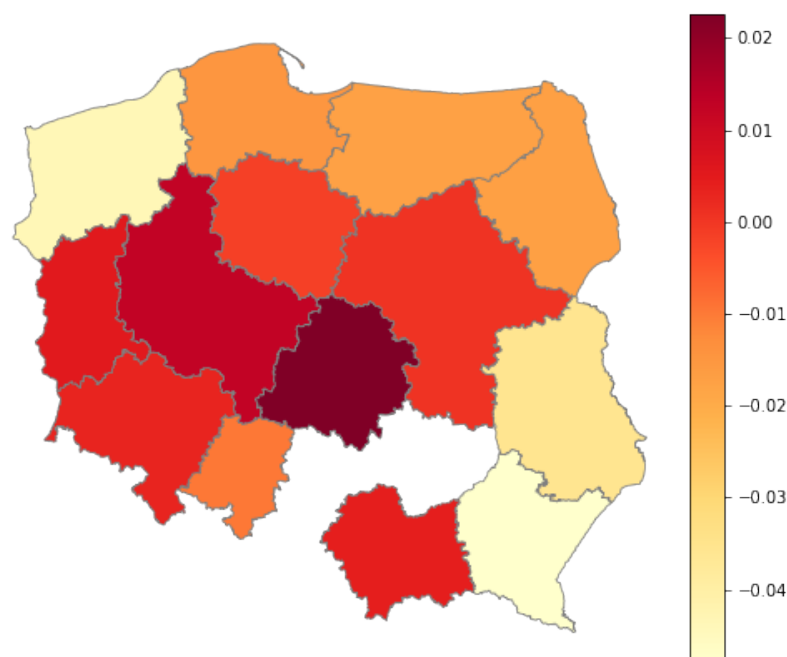


Rysunek 3.12. Współczynnik emisji dwutlenku węgla według województw wyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Prawie 30% całkowitej emisji CO₂ w UE pochodzi z sektora transportu, z czego 72% – z transportu drogowego. 38% emisji dwutlenku węgla transportu drogowego, wytwarzane jest właśnie przez samochody ciężarowe⁷. Na powyższym kartogramie widzimy jak rozkładają się współczynniki modelu dla emisji CO₂. Województwo łódzkie ponownie przoduje w estymacji (4.3582), a województwo zachodniopomorskie zamyka stawkę z wartością 4.2782. Ciekawym zjawiskiem jest to, że województwo małopolskie w powyższym rysunku przedstawione jest w ciemniejszym odcieniu niż w poprzednich modelach, co oznacza zwiększenie emisji dwutlenku węgla w tym województwie.

⁷<https://www.europarl.europa.eu/news/pl/headlines/society/20190313STO31218/emisje-co2-z-samochodow-fakty-i-liczby-infografika>



Rysunek 3.13. Współczynnik emisji dwutlenku węgla według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Nieuwzględnione zostały dwa województwa przyjezdne - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Niewszystkie współczynniki występujące w modelu są statystycznie istotne. Podczas wyliczania danego modelu, nieistotny okazał się parametr "to_mazowieckie", którego wartość p sięgnęła poziom aż 28%. W podziale na województwa przyjezdne, swój stosunek wobec innych województw zmieniło również województwo dolnośląskie, którego barwa na kartogramie zmieniła odcień na znacznie ciemniejszy.

Tabela 3.8. Oszacowania współczynników modelu dla emisji dwutlenku węgla

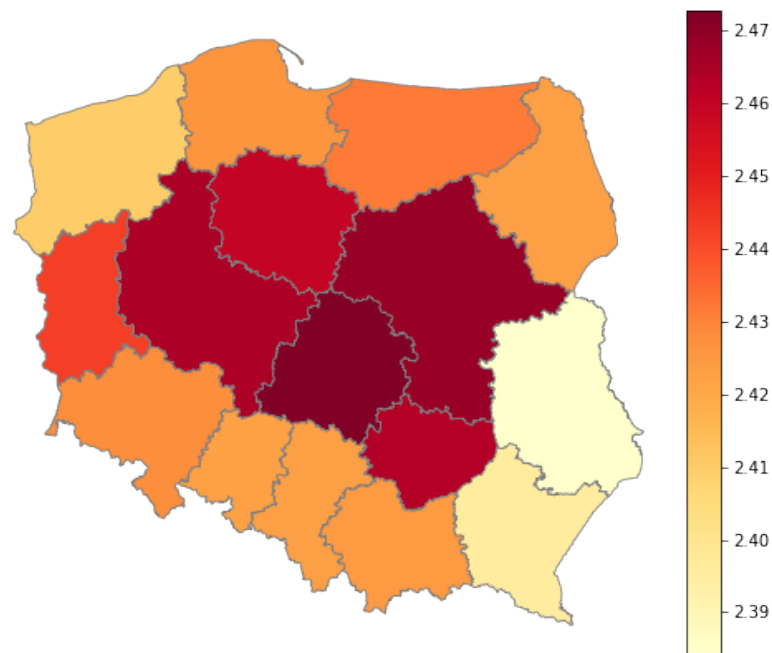
	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0033	3.22e-06	1031.811	0.000	0.003	0.003
time_h	-0.0221	5.44e-05	-406.101	0.000	-0.022	-0.022
toll_distance_km	-0.0004	2.97e-06	-145.158	0.000	-0.000	-0.000
toll_cost_euro	-0.0010	1.73e-05	-57.138	0.000	-0.001	-0.001
manoeuvres	0.0032	1.76e-05	179.683	0.000	0.003	0.003
from_dolnośląskie	4.3233	0.001	4342.637	0.000	4.321	4.325

Kontynuacja na następnej stronie

from_kujawsko-pomorskie	4.3261	0.001	4865.106	0.000	4.324	4.328
from_lubelskie	4.2836	0.001	4264.935	0.000	4.282	4.286
from_lubuskie	4.3271	0.001	4168.430	0.000	4.325	4.329
from_mazowieckie	4.3504	0.001	5408.992	0.000	4.349	4.352
from_małopolskie	4.3354	0.001	4110.486	0.000	4.333	4.337
from_opolskie	4.3055	0.001	3660.102	0.000	4.303	4.308
from_podkarpackie	4.2906	0.001	3421.379	0.000	4.288	4.293
from_podlaskie	4.3052	0.001	4313.158	0.000	4.303	4.307
from_pomorskie	4.2998	0.001	4371.397	0.000	4.298	4.302
from_warminsko-mazurskie	4.3055	0.001	4170.846	0.000	4.303	4.307
from_wielkopolskie	4.3352	0.001	5068.774	0.000	4.334	4.337
from_zachodniopomorskie	4.2782	0.001	4362.824	0.000	4.276	4.280
from_łódzkie	4.3582	0.001	5261.308	0.000	4.357	4.360
from_śląskie	4.3147	0.001	3978.297	0.000	4.313	4.317
from_świętokrzyskie	4.3507	0.001	4771.464	0.000	4.349	4.352
to_dolnośląskie	0.0031	0.001	5.507	0.000	0.002	0.004
to_kujawsko-pomorskie	-0.0015	0.001	-2.586	0.010	-0.003	-0.000
to_lubelskie	-0.0356	0.001	-61.720	0.000	-0.037	-0.034
to_lubuskie	0.0055	0.001	8.854	0.000	0.004	0.007
to_mazowieckie	0.0007	0.001	1.080	0.280	-0.001	0.002
to_małopolskie	0.0042	0.001	7.568	0.000	0.003	0.005
to_opolskie	-0.0096	0.001	-16.920	0.000	-0.011	-0.009
to_podkarpackie	-0.0479	0.001	-68.088	0.000	-0.049	-0.046
to_podlaskie	-0.0172	0.001	-30.491	0.000	-0.018	-0.016
to_pomorskie	-0.0146	0.001	-25.127	0.000	-0.016	-0.013
to_warminsko-mazurskie	-0.0174	0.001	-30.296	0.000	-0.018	-0.016
to_wielkopolskie	0.0125	0.001	21.569	0.000	0.011	0.014
to_zachodniopomorskie	-0.0437	0.001	-65.799	0.000	-0.045	-0.042
to_łódzkie	0.0226	0.001	34.131	0.000	0.021	0.024

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.6 Węglowodory

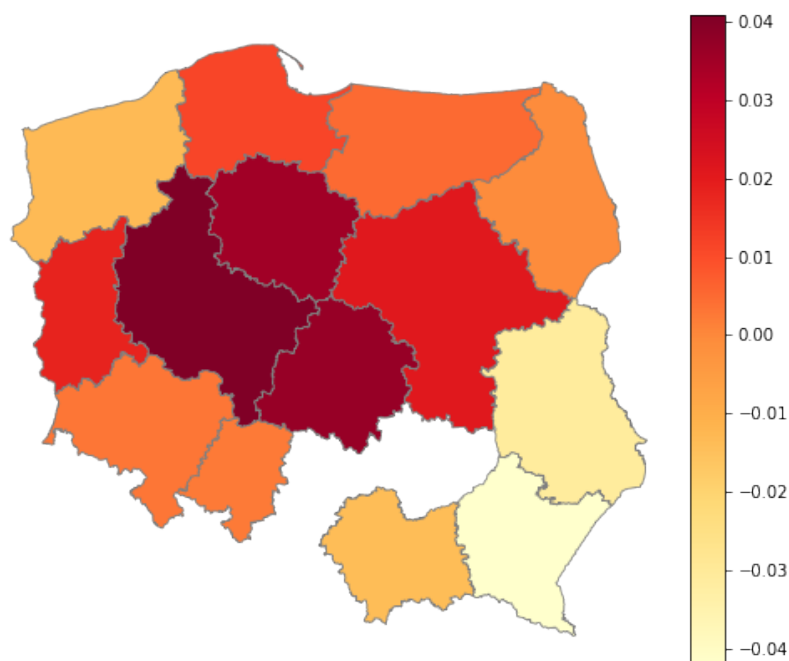


Rysunek 3.14. Współczynnik emisji węglowodorów według województw wyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Węglowodory są podstawowym składnikiem ropy naftowej, która stanowi ich podstawowe źródło w przemyśle⁸. Na powyższym kartogramie widzimy jak rozkładają się współczynniki modelu dla emisji węglowodorów. Największą wartość osiągnęło województwo łódzkie - 2.4728, a najmniejszą województwo lubelskie - 2.3842. Niewielka różnica występuje pomiędzy łódzkim, a mazowieckim, które parametr strukturalny został oszacowany na 2.4682.

⁸<https://pl.wikipedia.org/wiki/W%C4%99glowodory>



Rysunek 3.15. Współczynnik emisji węglowodorów według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Nieuwzględnione zostały dwa województwa przyjezdne - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Najbardziej podatnym na emisje węglowodorów przy kryterium województwa przyjezdnego, okazało się wielkopolskie. Podczas wyliczania danego modelu, nieistotny okazał się parametr "to_podlaskie", którego wartość p sięgnęła poziom aż 35%.

Tabela 3.9. Oszacowania współczynników modelu dla emisji węglowodorów

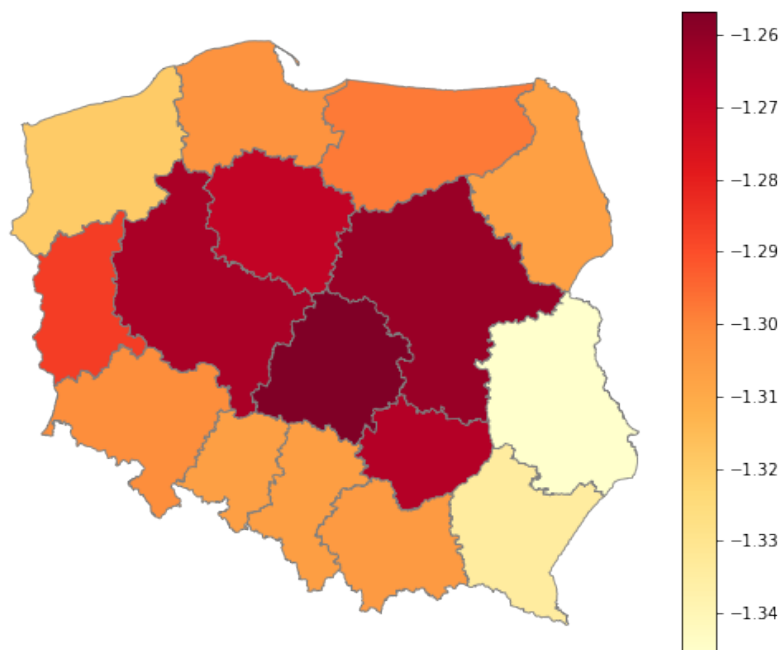
	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0031	3.24e-06	953.424	0.000	0.003	0.003
time_h	-0.0175	5.43e-05	-322.936	0.000	-0.018	-0.017
toll_distance_km	-0.0005	3.03e-06	-178.539	0.000	-0.001	-0.001
toll_cost_euro	-0.0004	1.68e-05	-23.706	0.000	-0.000	-0.000
manoeuvres	0.0045	1.81e-05	246.682	0.000	0.004	0.004
from_dolnośląskie	2.4283	0.001	2439.208	0.000	2.426	2.430
from_kujawsko-pomorskie	2.4600	0.001	2765.435	0.000	2.458	2.462
from_lubelskie	2.3842	0.001	2380.178	0.000	2.382	2.386

Kontynuacja na następnej stronie

from_lubuskie	2.4430	0.001	2374.884	0.000	2.441	2.445
from_mazowieckie	2.4682	0.001	3082.836	0.000	2.467	2.470
from_małopolskie	2.4244	0.001	2318.413	0.000	2.422	2.426
from_opolskie	2.4228	0.001	2092.156	0.000	2.421	2.425
from_podkarpackie	2.3956	0.001	1949.476	0.000	2.393	2.398
from_podlaskie	2.4230	0.001	2462.724	0.000	2.421	2.425
from_pomorskie	2.4263	0.001	2489.816	0.000	2.424	2.428
from_warminsko-mazurskie	2.4319	0.001	2359.358	0.000	2.430	2.434
from_wielkopolskie	2.4647	0.001	2920.613	0.000	2.463	2.466
from_zachodniopomorskie	2.4100	0.001	2491.049	0.000	2.408	2.412
from_łódzkie	2.4728	0.001	3009.775	0.000	2.471	2.474
from_śląskie	2.4233	0.001	2247.249	0.000	2.421	2.425
from_świętokrzyskie	2.4634	0.001	2714.506	0.000	2.462	2.465
to_dolnośląskie	0.0030	0.001	5.215	0.000	0.002	0.004
to_kujawsko-pomorskie	0.0351	0.001	58.862	0.000	0.034	0.036
to_lubelskie	-0.0308	0.001	-52.309	0.000	-0.032	-0.030
to_lubuskie	0.0181	0.001	29.561	0.000	0.017	0.019
to_mazowieckie	0.0209	0.001	32.188	0.000	0.020	0.022
to_małopolskie	-0.0136	0.001	-25.078	0.000	-0.015	-0.013
to_opolskie	0.0024	0.001	4.284	0.000	0.001	0.004
to_podkarpackie	-0.0421	0.001	-61.425	0.000	-0.043	-0.041
to_podlaskie	-0.0005	0.001	-0.928	0.353	-0.002	0.001
to_pomorskie	0.0113	0.001	18.888	0.000	0.010	0.013
to_warminsko-mazurskie	0.0050	0.001	8.293	0.000	0.004	0.006
to_wielkopolskie	0.0408	0.001	70.008	0.000	0.040	0.042
to_zachodniopomorskie	-0.0130	0.001	-19.444	0.000	-0.014	-0.012
to_łódzkie	0.0367	0.001	55.468	0.000	0.035	0.038

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.7 Metan

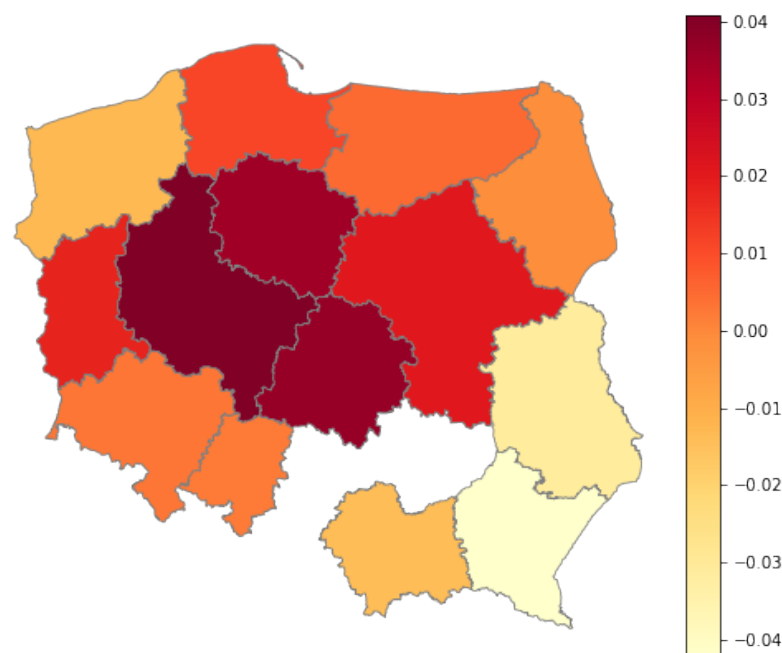


Rysunek 3.16. Współczynnik emisji metanu według województw wyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Metan jest drugim najpowszechniejszym gazem cieplarnianym. Cząsteczki tego gazu są w stanie zatrzymać w atmosferze aż 28 razy więcej ilości ciepła niż dwutlenek węgla, który przez wysokie stężenie w atmosferze jest źródłem największych zmian klimatycznych. Różnica między tymi gazami polega na tym, że po wyemitowaniu do atmosfery, metan utrzymuje się w niej 12 lat, w przeciwieństwie do dwutlenku węgla, który pozostaje w niej 100 lat. Na obecną chwilę szacowane jest, że transport drogowy jest źródłem 4% światowej emisji metanu⁹. Na rysunku 3.16 przedstawione są współczynniki dla emisji metanu według województw wyjezdnych. Tak jak na poprzednich kartogramach, najwyższe wskaźniki skupione są w centralnej Polsce. Największy współczynnik osiągnęło województwo łódzkie (-1.2567), a najmniejszy lubelskie (-1.3457).

⁹<https://przemyslisrodowisko.pl/metan-jak-ograniczyc-emisje/>



Rysunek 3.17. Współczynnik emisji metanu według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Nieuwzględnione zostały dwa województwa przyjezdne - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Najbardziej podatnym na emisje węglowodorów przy kryterium województwa przyjezdnego, okazało się wielkopolskie. Prawie wszystkie współczynniki występujące w modelu są statystycznie istotne. Podczas wyliczania danego modelu, nieistotny okazał się parametr "to_podlaskie", którego wartość p uzyskała poziom 13%.

Tabela 3.10. Oszacowania współczynników modelu dla emisji metanu

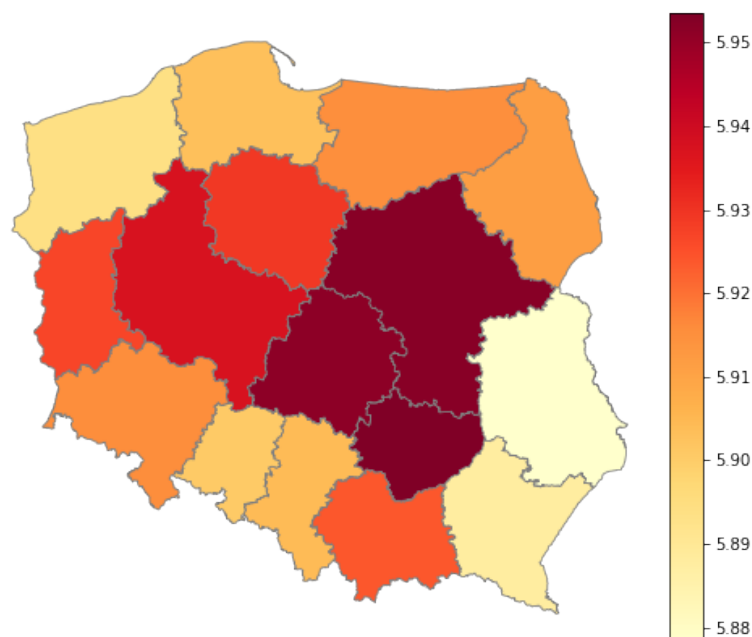
	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0031	3.23e-06	953.679	0.000	0.003	0.003
time_h	-0.0175	5.44e-05	-322.676	0.000	-0.018	-0.017
toll_distance_km	-0.0005	3.03e-06	-178.335	0.000	-0.001	-0.001
toll_cost_euro	-0.0004	1.68e-05	-23.465	0.000	-0.000	-0.000
manoeuvres	0.0045	1.81e-05	246.938	0.000	0.004	0.004
from_dolnośląskie	-1.3013	0.001	-1306.643	0.000	-1.303	-1.299
from_kujawsko-pomorskie	-1.2694	0.001	-1426.568	0.000	-1.271	-1.268
from_lubelskie	-1.3457	0.001	-1343.018	0.000	-1.348	-1.344

Kontynuacja na następnej stronie

from_lubuskie	-1.2865	0.001	-1250.367	0.000	-1.289	-1.285
from_mazowieckie	-1.2614	0.001	-1575.023	0.000	-1.263	-1.260
from_małopolskie	-1.3053	0.001	-1247.834	0.000	-1.307	-1.303
from_opolskie	-1.3067	0.001	-1128.032	0.000	-1.309	-1.304
from_podkarpackie	-1.3341	0.001	-1085.221	0.000	-1.337	-1.332
from_podlaskie	-1.3068	0.001	-1327.942	0.000	-1.309	-1.305
from_pomorskie	-1.3033	0.001	-1336.963	0.000	-1.305	-1.301
from_warminsko-mazurskie	-1.2978	0.001	-1258.822	0.000	-1.300	-1.296
from_wielkopolskie	-1.2647	0.001	-1498.192	0.000	-1.266	-1.263
from_zachodniopomorskie	-1.3196	0.001	-1363.638	0.000	-1.322	-1.318
from_łódzkie	-1.2567	0.001	-1529.112	0.000	-1.258	-1.255
from_śląskie	-1.3063	0.001	-1211.020	0.000	-1.308	-1.304
from_świętokrzyskie	-1.2662	0.001	-1394.592	0.000	-1.268	-1.264
to_dolnośląskie	0.0028	0.001	4.975	0.000	0.002	0.004
to_kujawsko-pomorskie	0.0351	0.001	58.957	0.000	0.034	0.036
to_lubelskie	-0.0313	0.001	-53.003	0.000	-0.032	-0.030
to_lubuskie	0.0181	0.001	29.496	0.000	0.017	0.019
to_mazowieckie	0.0208	0.001	31.899	0.000	0.019	0.022
to_małopolskie	-0.0138	0.001	-25.392	0.000	-0.015	-0.013
to_opolskie	0.0024	0.001	4.308	0.000	0.001	0.004
to_podkarpackie	-0.0424	0.001	-61.783	0.000	-0.044	-0.041
to_podlaskie	-0.0009	0.001	-1.512	0.131	-0.002	0.000
to_pomorskie	0.0113	0.001	18.780	0.000	0.010	0.012
to_warminsko-mazurskie	0.0048	0.001	7.901	0.000	0.004	0.006
to_wielkopolskie	0.0408	0.001	70.019	0.000	0.040	0.042
to_zachodniopomorskie	-0.0131	0.001	-19.548	0.000	-0.014	-0.012
to_łódzkie	0.0366	0.001	55.325	0.000	0.035	0.038

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.8 Tlenki azotu



Rysunek 3.18. Współczynnik emisji tlenków azotu według województw wyjezdnych

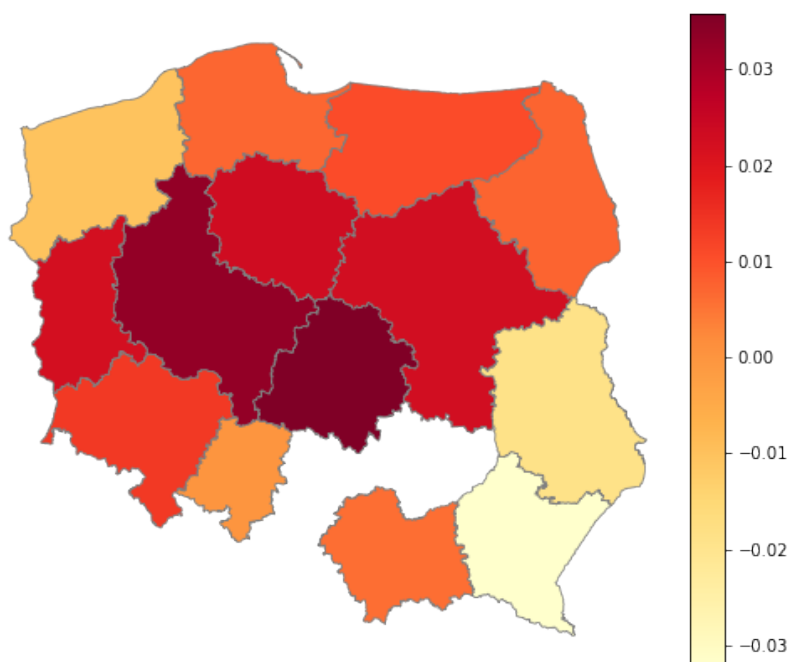
Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Równie ważne jak ograniczanie emisji dwutlenku siarki jest ograniczanie emisji tlenków azotu. Jest to grupa związków będących produktem spalania, którą jest niezwykle trudno wyeliminować ze spalin¹⁰. Tlenki azotu także należą do zanieczyszczeń, które są szkodliwe dla środowiska i zdrowia ludzkiego, mogą powodować lokalnie problemy zdrowotne oraz środowiskowe¹¹. Największa emisja NO₂ jest powodowana przez samochody z silnikiem Diesla¹². Dla utworzonego dla emisji tlenków azotu, modelu grawitacyjnego, w czołówce są województwo świętokrzyskie (5.9535), mazowieckie (5.9521) oraz łódzkie (5.9513). Różnice między nimi nie są aż tak duże. Najmniejszą wartość współczynnika emisji tlenków azotu osiągnęło, tak jak w poprzednich modelach, województwo lubelskie (5.8782). Warto zwrócić uwagę na województwo małopolskie, które zmieniło swoją barwę na ciemniejszą, w porównaniu do poprzednich modeli i oszacowań. W tym wypadku oznacza to zwiększoną emisję tlenków siarki względem pozostałych emisji.

¹⁰<https://elbelchatow.pgegiel.pl/Ochrona-srodowiska/Ograniczanie-emisji-tlenkow-azotu>

¹¹<https://www.gov.pl/web/gospodarkamorska/emisje-tlenkow-siarki-i-tlenkow-azotu>

¹²<https://airly.org/pl/tlenek-azotu-trujace-skladniki-smogu-cz-1/>



Rysunek 3.19. Współczynnik emisji tlenków azotu według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Nieuwzględnione zostały dwa województwa przyjezdne - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Prawie wszystkie współczynniki występujące w modelu są statystycznie istotne. Podczas wyliczania danego modelu, nieistotny okazał się parametr "to_opolskie", którego wartość p sięgnęła poziom aż 70%.

Tabela 3.11. Oszacowania współczynników modelu dla emisji tlenków azotu

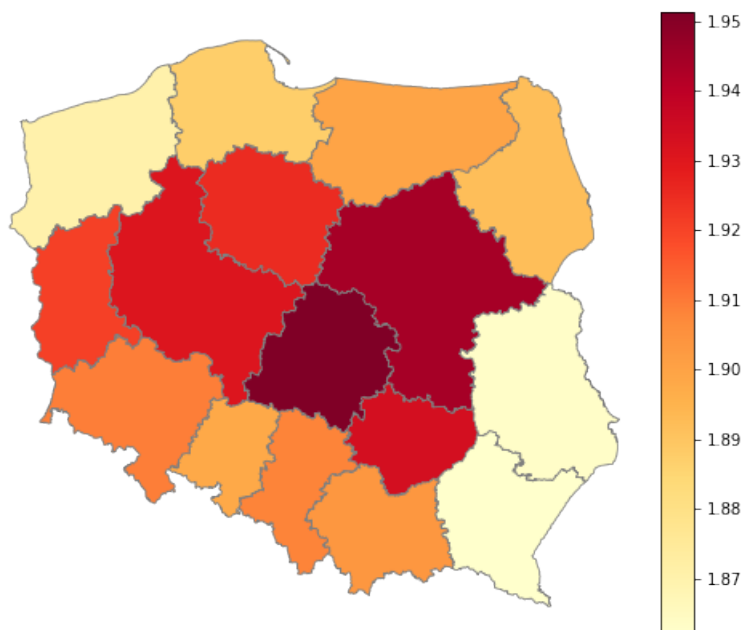
	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0032	3.44e-06	938.103	0.000	0.003	0.003
time_h	-0.0193	5.4e-05	-356.968	0.000	-0.019	-0.019
toll_distance_km	-0.0006	3.25e-06	-176.656	0.000	-0.001	-0.001
toll_cost_euro	-0.0010	1.67e-05	-57.821	0.000	-0.001	-0.001
manoeuvres	0.0039	1.85e-05	208.253	0.000	0.004	0.004
from_dolnośląskie	5.9158	0.001	6001.256	0.000	5.914	5.918
from_kujawsko-pomorskie	5.9292	0.001	6690.702	0.000	5.927	5.931
from_lubelskie	5.8782	0.001	5877.407	0.000	5.876	5.880

Kontynuacja na następnej stronie

from_lubuskie	5.9268	0.001	5773.400	0.000	5.925	5.929
from_mazowieckie	5.9521	0.001	7453.787	0.000	5.951	5.954
from_małopolskie	5.9240	0.001	5684.090	0.000	5.922	5.926
from_opolskie	5.9006	0.001	5100.093	0.000	5.898	5.903
from_podkarpackie	5.8876	0.001	4786.494	0.000	5.885	5.890
from_podlaskie	5.9117	0.001	5951.853	0.000	5.910	5.914
from_pomorskie	5.9030	0.001	6065.660	0.000	5.901	5.905
from_warminsko-mazurskie	5.9155	0.001	5699.445	0.000	5.914	5.918
from_wielkopolskie	5.9375	0.001	7025.326	0.000	5.936	5.939
from_zachodniopomorskie	5.8939	0.001	6055.729	0.000	5.892	5.896
from_łódzkie	5.9513	0.001	7260.217	0.000	5.950	5.953
from_śląskie	5.9043	0.001	5504.836	0.000	5.902	5.906
from_świętokrzyskie	5.9535	0.001	6633.906	0.000	5.952	5.955
to_dolnośląskie	0.0137	0.001	23.705	0.000	0.013	0.015
to_kujawsko-pomorskie	0.0232	0.001	39.910	0.000	0.022	0.024
to_lubelskie	-0.0187	0.001	-32.537	0.000	-0.020	-0.018
to_lubuskie	0.0224	0.001	36.001	0.000	0.021	0.024
to_mazowieckie	0.0227	0.001	35.617	0.000	0.021	0.024
to_małopolskie	0.0061	0.001	11.137	0.000	0.005	0.007
to_opolskie	0.0002	0.001	0.386	0.700	-0.001	0.001
to_podkarpackie	-0.0321	0.001	-46.810	0.000	-0.033	-0.031
to_podlaskie	0.0074	0.001	13.264	0.000	0.006	0.009
to_pomorskie	0.0070	0.001	11.947	0.000	0.006	0.008
to_warminsko-mazurskie	0.0109	0.001	18.680	0.000	0.010	0.012
to_wielkopolskie	0.0330	0.001	57.240	0.000	0.032	0.034
to_zachodniopomorskie	-0.0103	0.001	-15.665	0.000	-0.012	-0.009
to_łódzkie	0.0358	0.001	54.436	0.000	0.035	0.037

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.9 Podtlenek azotu



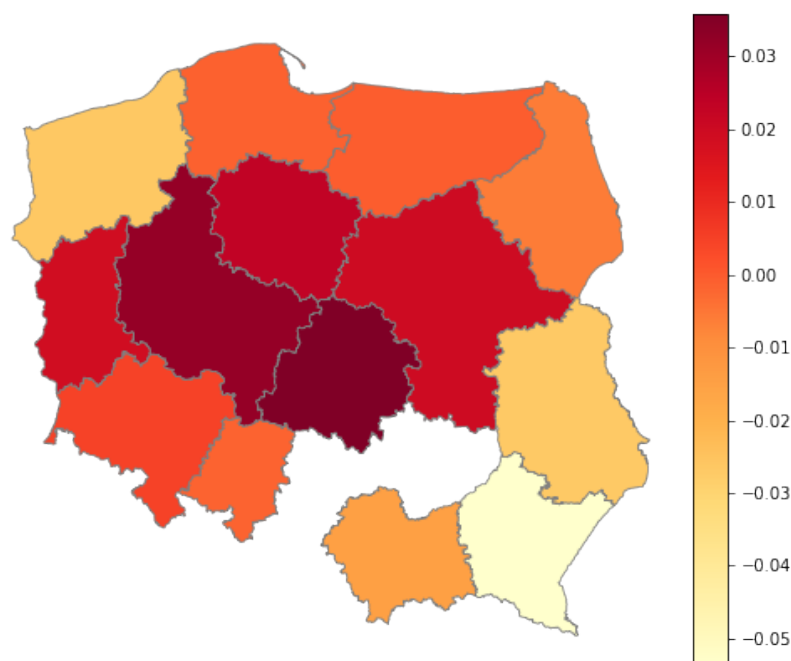
Rysunek 3.20. Współczynnik emisji podtlenu azotu według województw wyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Podtlenu azotu jest już w atmosferze o 20 proc. więcej w stosunku do poziomu sprzed epoki przemysłowej. Wzrost stężenia tego gazu cieplarnianego jest spowodowany rosnącym zużyciem nawozów azotowych w światowej produkcji żywności¹³. "Podtlenek azotu zajmuje obecnie trzecie miejsce w rankingu gazów, które przyczyniły się do wzrostu średniej temperatury powierzchni Ziemi. Jego koncentracje wzrosły w mniejszym stopniu niż stężenia dwutlenku węgla i metanu. Nie oznacza to jednak, że można go lekceważyć. Jest go w atmosferze stosunkowo mało i w związku z tym nawet niewielki wzrost koncentracji przekłada się na istotną zmianę w bilansie energetycznym planety"¹⁴. Dla emisji podtlenu azotu, największy współczynnik osiągnęło województwo łódzkie (1.9512), a najmniejszą wartość ma województwo podkarpackie (1.8620). Jest to pierwszy pierwiastek, dla którego województwo podkarpackie osiągnęło wartość najmniejszą dla danego modelu. Niewiele więcej ma często wymieniane województwo lubelskie, które różni się od podkarpackiego o 0.0013 jednostek - 1.8633.

¹³<https://naukawpolsce.pap.pl/aktualnosci/news%2C84244%2Cbadanie-w-atmosferze-rosnie-steze-nie-podtlenu-azotu.html>

¹⁴<https://naukawpolsce.pap.pl/aktualnosci/news%2C84244%2Cbadanie-w-atmosferze-rosnie-steze-nie-podtlenu-azotu.html>



Rysunek 3.21. Współczynnik emisji podtlenku azotu według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Nieuwzględnione zostały dwa województwa przyjezdne - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Prawie wszystkie współczynniki występujące w modelu są statystycznie istotne. Podczas wyliczania danego modelu, nieistotny okazał się parametr "to_warminsko-mazurskie", którego wartość p sięgnęła poziom aż 49% oraz "to_pomorskie", którego wartość p ma ponad 5%.

Tabela 3.12. Oszacowania współczynników modelu dla emisji podtlenku azotu

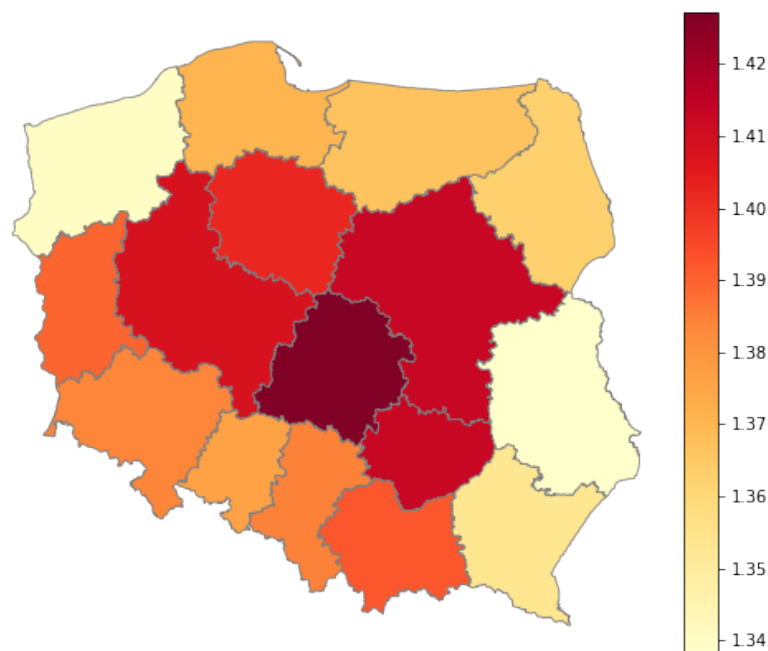
	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0034	3.43e-06	991.491	0.000	0.003	0.003
time_h	-0.0225	5.56e-05	-404.839	0.000	-0.023	-0.022
toll_distance_km	-0.0005	3.2e-06	-158.826	0.000	-0.001	-0.001
toll_cost_euro	-0.0009	1.77e-05	-49.286	0.000	-0.001	-0.001
manoeuvres	0.0029	1.84e-05	159.693	0.000	0.003	0.003
from_dolnośląskie	1.9091	0.001	1868.963	0.000	1.907	1.911
from_kujawsko-pomorskie	1.9252	0.001	2106.166	0.000	1.923	1.927

Kontynuacja na następnej stronie

from_lubelskie	1.8633	0.001	1795.825	0.000	1.861	1.865
from_lubuskie	1.9206	0.001	1806.245	0.000	1.919	1.923
from_mazowieckie	1.9442	0.001	2361.324	0.000	1.943	1.946
from_małopolskie	1.9035	0.001	1747.863	0.000	1.901	1.906
from_opolskie	1.8982	0.001	1590.027	0.000	1.896	1.901
from_podkarpackie	1.8620	0.001	1442.091	0.000	1.859	1.865
from_podlaskie	1.8920	0.001	1836.236	0.000	1.890	1.894
from_pomorskie	1.8877	0.001	1871.299	0.000	1.886	1.890
from_warminsko-mazurskie	1.8997	0.001	1783.343	0.000	1.898	1.902
from_wielkopolskie	1.9308	0.001	2203.758	0.000	1.929	1.933
from_zachodniopomorskie	1.8704	0.001	1856.122	0.000	1.868	1.872
from_łódzkie	1.9512	0.001	2304.134	0.000	1.950	1.953
from_śląskie	1.9082	0.001	1719.646	0.000	1.906	1.910
from_świętokrzyskie	1.9336	0.001	2072.731	0.000	1.932	1.935
to_dolnośląskie	0.0049	0.001	8.533	0.000	0.004	0.006
to_kujawsko-pomorskie	0.0236	0.001	39.612	0.000	0.022	0.025
to_lubelskie	-0.0266	0.001	-44.687	0.000	-0.028	-0.025
to_lubuskie	0.0190	0.001	29.929	0.000	0.018	0.020
to_mazowieckie	0.0201	0.001	30.945	0.000	0.019	0.021
to_małopolskie	-0.0148	0.001	-26.700	0.000	-0.016	-0.014
to_opolskie	-0.0013	0.001	-2.282	0.022	-0.002	-0.000
to_podkarpackie	-0.0537	0.001	-75.035	0.000	-0.055	-0.052
to_podlaskie	-0.0060	0.001	-10.340	0.000	-0.007	-0.005
to_pomorskie	-0.0011	0.001	-1.895	0.058	-0.002	3.86e-05
to_warminsko-mazurskie	-0.0004	0.001	-0.688	0.492	-0.002	0.001
to_wielkopolskie	0.0316	0.001	53.219	0.000	0.030	0.033
to_zachodniopomorskie	-0.0264	0.001	-38.853	0.000	-0.028	-0.025
to_łódzkie	0.0358	0.001	52.942	0.000	0.035	0.037

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.10 Cząstki stałe



Rysunek 3.22. Współczynnik emisji cząstek stałych według województw wyjezdnych

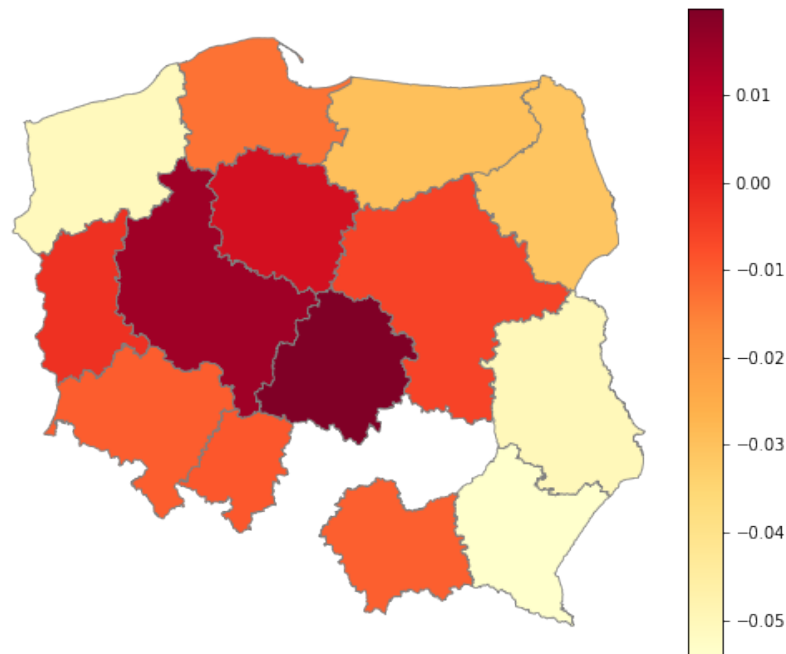
Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Na podstawie definicji z książki poznańskich naukowców politechniki poznańskiej (Merkisz i Pielecha, 2024) pojęcie cząstek stałych jest niejednoznaczne, gdyż:

- nie można ich jednoznacznie zdefiniować ani fizycznie, ani chemicznie,
- są większe niż molekuły gazów,
- nie są jednakowe pod względem wymiarów i kształtów,
- są nieregularną mieszaniną składników chemicznych,
- mogą być odseparowane (odfiltrowane, a nie odsiane) za pomocą filtra (pory większe od rozmiarów PM).

Cząstki stałe są bardziej charakterystyczne dla pojazdów zasilanych olejem napędowym, jako że w przeszłości (zanim wprowadzono czyste silniki wysokoprężne z filtrami cząstek stałych (DPF) silniki Diesla produkowały więcej cząstek stałych niż silniki benzynowe. W ujęciu emisji cząstek stałych, największą wartość osiągnęło województwo

łódzkie (1.4272), a najmniejsze lubelskie. Rozkład między województwami przy emisji cząstek stałych jest bardzo podobny to większości poprzednich modeli, które były opisywane.



Rysunek 3.23. Współczynnik emisji cząstek stałych według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Nieuwzględnione zostały dwa województwa przyjezdne - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Wszystkie współczynniki występujące w modelu są statystycznie istotne, gdyż żaden z nich nie osiągnął wartości p większej niż 5%.

Tabela 3.13. Oszacowania współczynników modelu dla emisji cząstek stałych

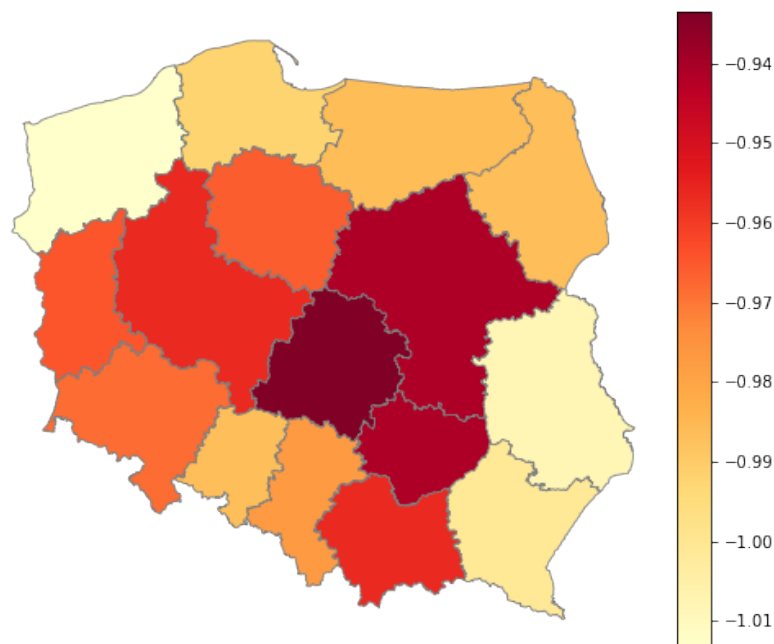
	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0032	3e-06	1062.853	0.000	0.003	0.003
time_h	-0.0205	5.38e-05	-380.595	0.000	-0.021	-0.020
toll_distance_km	-0.0004	2.73e-06	-147.776	0.000	-0.000	-0.000
toll_cost_euro	-0.0004	1.7e-05	-22.601	0.000	-0.000	-0.000
manoeuvres	0.0038	1.69e-05	224.686	0.000	0.004	0.004
from_dolnośląskie	1.3836	0.001	1395.512	0.000	1.382	1.386

Kontynuacja na następnej stronie

from_kujawsko-pomorskie	1.4020	0.001	1598.715	0.000	1.400	1.404
from_lubelskie	1.3380	0.001	1352.343	0.000	1.336	1.340
from_lubuskie	1.3896	0.001	1360.845	0.000	1.388	1.392
from_mazowieckie	1.4132	0.001	1782.422	0.000	1.412	1.415
from_małopolskie	1.3920	0.001	1334.028	0.000	1.390	1.394
from_opolskie	1.3761	0.001	1182.670	0.000	1.374	1.378
from_podkarpackie	1.3531	0.001	1094.016	0.000	1.351	1.356
from_podlaskie	1.3624	0.001	1400.311	0.000	1.360	1.364
from_pomorskie	1.3707	0.001	1416.343	0.000	1.369	1.373
from_warminsko-mazurskie	1.3666	0.001	1357.372	0.000	1.365	1.369
from_wielkopolskie	1.4080	0.001	1677.269	0.000	1.406	1.410
from_zachodniopomorskie	1.3402	0.001	1398.408	0.000	1.338	1.342
from_łódzkie	1.4272	0.001	1744.652	0.000	1.426	1.429
from_śląskie	1.3845	0.001	1286.182	0.000	1.382	1.387
from_świętokrzyskie	1.4126	0.001	1568.202	0.000	1.411	1.414
to_dolnośląskie	-0.0100	0.001	-18.324	0.000	-0.011	-0.009
to_kujawsko-pomorskie	0.0053	0.001	9.107	0.000	0.004	0.006
to_lubelskie	-0.0502	0.001	-87.357	0.000	-0.051	-0.049
to_lubuskie	-0.0027	0.001	-4.485	0.000	-0.004	-0.002
to_mazowieckie	-0.0059	0.001	-9.346	0.000	-0.007	-0.005
to_małopolskie	-0.0103	0.001	-18.908	0.000	-0.011	-0.009
to_opolskie	-0.0091	0.001	-16.499	0.000	-0.010	-0.008
to_podkarpackie	-0.0541	0.001	-78.272	0.000	-0.055	-0.053
to_podlaskie	-0.0309	0.001	-55.192	0.000	-0.032	-0.030
to_pomorskie	-0.0130	0.001	-22.428	0.000	-0.014	-0.012
to_warminsko-mazurskie	-0.0298	0.001	-51.857	0.000	-0.031	-0.029
to_wielkopolskie	0.0153	0.001	26.673	0.000	0.014	0.016
to_zachodniopomorskie	-0.0509	0.001	-77.506	0.000	-0.052	-0.050
to_łódzkie	0.0199	0.001	30.524	0.000	0.019	0.021

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

3.2.11 Dwutlenek siarki



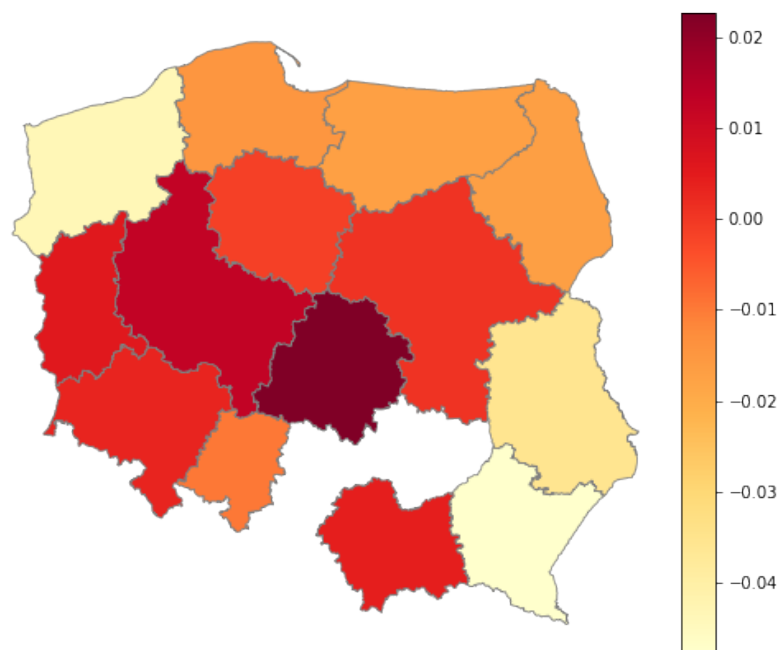
Rysunek 3.24. Współczynnik emisji dwutlenku siarki według województw wyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Siarczyny stanowią istotny składnik kwaśnych deszczów czy smogu, z którym zmagają się mieszkańcy dużych miast¹⁵. Ropa, która jest podstawowym paliwem wśród samochodów ciężarowych, zawiera różne ilości siarki, w zależności od źródła, ale zwykle pomiędzy 1% a 5%. W trakcie procesu spalania, siarka zawarta w paliwie przetwarzana jest prawie w całości na dwutlenek siarki. Ten stanowi także składnik smogu w wielkich aglomeracjach miejskich¹⁶. Tutaj również największy współczynnik emisji SO_2 osiągnęło województwo łódzkie z wynikiem -0.9335, a najmniejszy zachodniopomorskie (-1.0134), które dawno nie było tak nisko w klasyfikacji. Swoją kolor zaciemniło (zwiększyło wartość) województwo małopolskie, które poprzednio również miała większe wyliczenia dla poszczególnych pierwiastków.

¹⁵<https://nuscana.pl/dwutlenek-siarki-so2-jednak-szkodzi/>

¹⁶<https://edroga.pl/ochrona-srodowiska/oddziaływanie-transportu-drogowego-na-jakosc-powietrza-atmosferycznego-cz-2-111211252>



Rysunek 3.25. Współczynnik emisji dwutlenku siarki według województw przyjezdnych

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Nieuwzględnione zostały dwa województwa przyjezdne - świętokrzyskie oraz śląskie. Województwo świętokrzyskie zostało wykluczone na podstawie warunku zerowych przepływów, a śląskie zostało zakwalifikowane jako zmienna idealnie współliniowa. Podczas wyliczania danego modelu, nieistotny okazał się parametr "to_mazowieckie", którego wartość p sięgnęła poziom 19%.

Tabela 3.14. Oszacowania współczynników modelu dla emisji dwutlenku siarki

	coef	std err	t	P> t	[0.025	0.975]
distance_km	0.0033	3.22e-06	1031.284	0.000	0.003	0.003
time_h	-0.0221	5.44e-05	-406.293	0.000	-0.022	-0.022
toll_distance_km	-0.0004	2.98e-06	-144.997	0.000	-0.000	-0.000
toll_cost_euro	-0.0010	1.73e-05	-57.760	0.000	-0.001	-0.001
manoeuvres	0.0032	1.76e-05	179.430	0.000	0.003	0.003
from_dolnośląskie	-0.9683	0.001	-972.781	0.000	-0.970	-0.966
from_kujawsko-pomorskie	-0.9658	0.001	-1086.084	0.000	-0.968	-0.964
from_lubelskie	-1.0081	0.001	-1003.671	0.000	-1.010	-1.006
from_lubuskie	-0.9645	0.001	-929.142	0.000	-0.967	-0.962
from_mazowieckie	-0.9413	0.001	-1170.310	0.000	-0.943	-0.940

Kontynuacja na następnej stronie

from_małopolskie	-0.9563	0.001	-906.783	0.000	-0.958	-0.954
from_opolskie	-0.9863	0.001	-838.440	0.000	-0.989	-0.984
from_podkarpackie	-1.0012	0.001	-798.407	0.000	-1.004	-0.999
from_podlaskie	-0.9864	0.001	-988.115	0.000	-0.988	-0.984
from_pomorskie	-0.9920	0.001	-1008.480	0.000	-0.994	-0.990
from_warminsko-mazurskie	-0.9862	0.001	-955.172	0.000	-0.988	-0.984
from_wielkopolskie	-0.9566	0.001	-1118.394	0.000	-0.958	-0.955
from_zachodniopomorskie	-1.0134	0.001	-1033.307	0.000	-1.015	-1.011
from_łódzkie	-0.9335	0.001	-1127.019	0.000	-0.935	-0.932
from_śląskie	-0.9772	0.001	-901.073	0.000	-0.979	-0.975
from_świętokrzyskie	-0.9410	0.001	-1032.073	0.000	-0.943	-0.939
to_dolnośląskie	0.0034	0.001	6.004	0.000	0.002	0.004
to_kujawsko-pomorskie	-0.0015	0.001	-2.607	0.009	-0.003	-0.000
to_lubelskie	-0.0354	0.001	-61.402	0.000	-0.037	-0.034
to_lubuskie	0.0057	0.001	9.172	0.000	0.004	0.007
to_mazowieckie	0.0008	0.001	1.284	0.199	-0.000	0.002
to_małopolskie	0.0043	0.001	7.817	0.000	0.003	0.005
to_opolskie	-0.0096	0.001	-16.811	0.000	-0.011	-0.008
to_podkarpackie	-0.0478	0.001	-68.004	0.000	-0.049	-0.046
to_podlaskie	-0.0169	0.001	-30.040	0.000	-0.018	-0.016
to_pomorskie	-0.0146	0.001	-25.097	0.000	-0.016	-0.013
to_warminsko-mazurskie	-0.0172	0.001	-29.952	0.000	-0.018	-0.016
to_wielkopolskie	0.0126	0.001	21.723	0.000	0.011	0.014
to_zachodniopomorskie	-0.0434	0.001	-65.421	0.000	-0.045	-0.042
to_łódzkie	0.0227	0.001	34.275	0.000	0.021	0.024

Źródło: Opracowanie własne w języku Python na podstawie zebranych danych

Zakończenie

Najważniejszym aspektem niniejszej pracy było wykorzystanie modeli ciążenia w celu zbadania, czy w danym regionie występują tendencje do generowania wyższych/nieższych emisji podczas transportu drogowego. Praca nie wyczerpuje do końca tematu, jednak pozwala przedstawić istotne informacje dla całego społeczeństwa. Daje również szersze spojrzenie na dbałość o środowisko oraz transport drogowy samochodów ciężarowych w dzisiejszych czasach.

Celem pracy było stworzenie wizualizacji oraz wyznaczenie głównych statystyk związanych z emisją spalin w transporcie drogowym. Znaczącą część pracy pochłonęło przygotowanie oraz wygenerowanie danych, natomiast kluczowym elementem było przeprowadzenie obliczeń związanych z modelem grawitacyjnym, jego analizą oraz zastosowaniem w kontekście emisji spalin w logistyce.

W pracy udało się uzyskać bardzo ciekawe wnioski oraz wizualizacje, pozwalające na lepsze zauważenie pewnych trendów i zależności. Ważne było również przygotowanie danych do badania, co pozwoliło na utworzenie kompletnej i wiarygodnej bazy danych kodów pocztowych w Polsce wraz z ich reprezentacją w układzie współrzędnych geograficznych. Ciekawym spostrzeżeniem jest to, że emisyjność poszczególnych pierwiastków nie jest ze sobą ściśle powiązana. Województwo może mieć większy współczynnik dla wytwarzania dwutlenku siarki, ale już inny dla emisji cząstek stałych pod względem pozostałych województw i relacji między nimi. Największe wartości współczynników wydalania spalin wraz z zawartymi w nich zanieczyszczeniami skupione są w centralnej Polsce. Mogło mieć to ściśle powiązanie z zaawansowaniem infrastruktury drogowej, gęstością zaludnienia oraz liczby mieszkańców konkretnych województw. Najbardziej wyróżniającym się województwem okazało się województwo łódzkie, które pod prawie każdym względem, przodowało w oszacowaniach parametrów strukturalnych modelu. Z drugiej strony częstymi wskazaniem na najniższe wartości parametrów, wyróżniały

się województwo zachodniopomorskie oraz lubelskie. Utworzone kartogramy, wykresy oraz analizy ułatwiają zrozumienie wygenerowanych wyników oraz nakłaniają do głębszych refleksji oraz szukania bardziej złożonych zależności.

Wyniki badań są bardzo ciekawe i przedstawiają różne perspektywy emisyjności poszczególnych pierwiastków. Jest tu również szerokie pole do dalszych analiz z użyciem modelu grawitacyjnego.

Podsumowując warto zauważyć, że transport i logistyka jest bardzo ważnym aspektem gospodarczym każdego kraju. Wraz ze wzrostem natężenia ruchu drogowego pomiędzy miastami całego świata, rośnie również ilość szkodliwych pierwiastków wydanych do atmosfery. Praca ta może być ciekawym przedstawieniem mniej oczywistych przyczyn wpływających na zanieczyszczenia wynikające z transportu drogowego. Skoro coraz bardziej zaczynamy się troszczyć o środowisko i szukamy nowych, drogich rozwiązań to równie dobrze, możemy się skupić na aspektach niewymagających wkładu finansowego, a dobrego zarządzania flotą, logistyką oraz wszystkimi obszarami z nimi związanymi.

Bibliografia

- Ali, E. (2020). Geographic Information System (GIS): Definition, Development, Applications Components.
- Bartosik, Piątek & Loska. (2015). TSL SERVICES MARKET AS A SUPPORT SYSTEM OF THE POLISH ECONOMY.
- Borowczyk & Kaczmarek. (2007). Obliczanie bieżących współrzędnych celu powietrznego w oparciu o współrzędne mierzone przez odległe urządzenia śledzące.
- Enders, F. B. (2013). Collinearity.
- Heffner & Gibas. (2007). *Analiza ekonomiczno-przestrzenna*. Wydawnictwo Akademii Ekonomicznej w Katowicach.
- Herman, P. (2021). gegravity: General Equilibrium Gravity Modeling in Python.
- Huisman & Rolf. (2009). *Principles of Geographic Information Systems*. The International Institute for Geo-Information Science; Earth Observation.
- Klimczak, Ł. (2015). Model grawitacyjny jako narzędzie analizy handlu zagranicznego.
- Lima, M. (2017). Isaac Newton - Main contributions to the development of fluid mechanics.
- Mendie, P. (2016). Isaac Newton.
- Merkisz, J. & Pielecha, J. (2024). *Emisja cząstek stałych ze źródeł motoryzacyjnych*. Wydawnictwo Politechniki Poznańskiej.
- Pietrzak, M., Wilk, J. & Matusik, S. (2013). Analiza migracji wewnętrznych w Polsce z wykorzystaniem modelu grawitacji.
- Popowych, A. (2018). Zastosowanie modelu grawitacyjnego do analizy międzynarodowego handlu miodem pszczelim.
- Prasetya, Nguyen, Faizullin, Iswanto & Armay. (2019). Resolving the Shortest Path Problem using the Haversine Algorithm.
- Sagan, L. (2007). Delphi: pierwszy czujnik amoniaku do samochodu.

Słowiński, B. (2008). *Wprowadzenie do logistik*. Wydawnictwo Uczelniane Politechniki Koszalińskiej.

Sucheckie, B. (2010). *Ekonometria przestrzenna*. Wydawnictwo C.H. Beck.

Spis tabel

1.1	Trasy wygenerowane dla 00-002, Warszawa	15
3.1	Liczba utworzonych tras dla danego województwa	24
3.2	Statystyki opisowe wygenerowanych tras	26
3.3	Średnie wartości parametrów modelu z podziałem na województwa . . .	28
3.4	Oszacowania współczynników modelu dla zużycia paliwa	32
3.5	Oszacowania współczynników modelu dla emisji amoniaku	34
3.6	Oszacowania współczynników modelu dla emisji benzenu	37
3.7	Oszacowania współczynników modelu dla emisji tlenku węgla	40
3.8	Oszacowania współczynników modelu dla emisji dwutlenku węgla . . .	43
3.9	Oszacowania współczynników modelu dla emisji węglowodorów	46
3.10	Oszacowania współczynników modelu dla emisji metanu	49
3.11	Oszacowania współczynników modelu dla emisji tlenków azotu	52
3.12	Oszacowania współczynników modelu dla emisji podtlenku azotu	55
3.13	Oszacowania współczynników modelu dla emisji cząstek stałych	58
3.14	Oszacowania współczynników modelu dla emisji dwutlenku siarki	61

Spis rysunków

1.1	Wyznaczanie środka punktów	7
1.2	Liczba brakujących adresów w poszczególnych źródłach danych	8
1.3	Porównanie źródeł geokoordinat	13
1.4	Różnice odległości relacji Google-PTV	14
1.5	Średnia odległość pomiędzy punktami w danym województwie	14
3.1	Liczba tras wyjazdowych według województw w Polsce	24
3.2	Trasy dla 75-360, Koszalin	25
3.3	Średnie wartości poszczególnych parametrów modelu w podziale na województwa	29
3.4	Współczynnik spalania paliwa według województw wyjezdnych	30
3.5	Współczynnik spalania paliwa według województw przyjezdnych	31
3.6	Współczynnik emisji amoniaku według województw wyjezdnych	33
3.7	Współczynnik emisji amoniaku według województw przyjezdnych	34
3.8	Współczynnik emisji benzenu według województw wyjezdnych	36
3.9	Współczynnik emisji benzenu według województw przyjezdnych	37
3.10	Współczynnik emisji tlenu węgla według województw wyjezdnych	39
3.11	Współczynnik emisji tlenu węgla według województw przyjezdnych	40
3.12	Współczynnik emisji dwutlenku węgla według województw wyjezdnych	42
3.13	Współczynnik emisji dwutlenku węgla według województw przyjezdnych	43
3.14	Współczynnik emisji węglowodorów według województw wyjezdnych	45
3.15	Współczynnik emisji węglowodorów według województw przyjezdnych	46
3.16	Współczynnik emisji metanu według województw wyjezdnych	48
3.17	Współczynnik emisji metanu według województw przyjezdnych	49
3.18	Współczynnik emisji tlenków azotu według województw wyjezdnych	51
3.19	Współczynnik emisji tlenków azotu według województw przyjezdnych	52

3.20 Współczynnik emisji podtlenku azotu według województw wyjezdnych .	54
3.21 Współczynnik emisji podtlenku azotu według województw przyjezdnych	55
3.22 Współczynnik emisji cząstek stałych według województw wyjezdnych . .	57
3.23 Współczynnik emisji cząstek stałych według województw przyjezdnych .	58
3.24 Współczynnik emisji dwutlenku siarki według województw wyjezdnych .	60
3.25 Współczynnik emisji dwutlenku siarki według województw przyjezdnych	61

Kody języka Python

```
import numpy as np
import pandas as pd
import pyodbc
import json
import requests
from tqdm.notebook import tqdm, trange
conn = pyodbc.connect('Driver={SQL Server};'
                      'Server={SERVER_NAME};'
                      'Database={SERVER_DATABASE};'
                      'UID={SERVER_UID};'
                      'PWD={SERVER_PWD};')

cursor = conn.cursor()

def http_request_string(point_1_x, point_1_y, point_2_x, point_2_y):
    return '''{
        "waypoints": [
            {
                "$type": "TourPointDesc",
                "linkType": "NEXT_SEGMENT",
                "servicePeriod": 100,
                "useServicePeriodForRecreation": false,
                "completeServiceInIntervals": true,
                "openingIntervals": [],
                "coords": [
                    {
                        "point": {
                            "x": ''' + str(point_1_x) + ''',
                            "y": ''' + str(point_1_y) + ''',
                        }
                    }
                ]
            },
            {
                "$type": "TourPointDesc",
                "linkType": "NEXT_SEGMENT",
                "servicePeriod": 300,
                "useServicePeriodForRecreation": false,
                "completeServiceInIntervals": true,
                "openingIntervals": [],
                "coords": [
                    {
                        "point": {
                            "x": ''' + str(point_2_x) + ''',
                            "y": ''' + str(point_2_y) + ''',
                        }
                    }
                ]
            }
        ]
    },
    {
        "tourOptions": {},
        "routingOptions": [{
            "parameter": "OPTIMIZATION",
            "value": "90"
        }]
    }
    ]}]'''
```

```

}],
"exceptionPaths": [],
"details": {
  "cenEmissionConfiguration": {
    "$type": "CENEmissionConfiguration",
    "fleetSpecificAverageFuelConsumption": "32.2",
    "cenVersion": "CEN_2012"
  },
  "hbefaType": {
    "version": "HBEFA_3_2"
  },
  "emissions": {
    "$type": "EmissionType",
    "emissionLevel": "BASIC"
  },
  "manoeuvres": true,
  "texts": false,
  "nodes": false,
  "binaryPathDesc": false,
  "polygon": false,
  "detailLevel": "STANDARD",
  "dynamicInfo": false,
  "segments": false,
  "segmentAttributes": false,
  "manoeuvreAttributes": false,
  "brunnelManoeuvres": false,
  "urbanManoeuvres": false,
  "manoeuvreGroups": false,
  "manoeuvreGroupRatio": 1,
  "tollManoeuvres": false,
  "boundingRectanglesC": 5,
  "boundingRectanglesOffset": 300,
  "totalRectangle": false
},
"countryInfoOptions": {
  "detailedTollCosts": true,
  "allEuro": true,
  "calculatePartTollCosts": true
},
"callerContext": {
  "properties": [
    {
      "key": "ResponseGeometry",
      "value": "PLAIN"
    },
    {
      "key": "CoordFormat",
      "value": "OG_GEODECIMAL"
    },
    {
      "key": "Profile",
      "value": "truck40t_01-00_S"
    }
  ],
  {
    "key": "ProfileXMLSnippet",
    "value": "<?xml version='1.0' encoding='UTF-8'?><Profile xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' parent='truckfast'><Routing majorVersion='2' minorVersion='0'><Vehicle><Physical><Drive driveType='MOTORIZED'><Engine bioFuelRatio='0' fuelConsumption='32.2' fuelType='DIESEL' /></Drive></Physical></Vehicle><Course><AdditionalDataRules enabled='true'><VehicleSpecific enabled='true' /></AdditionalDataRules></Course></Routing><FeatureLayer majorVersion='1' minorVersion='0'><Themes><Theme id='PTV_TruckAttributes' enabled='true' /></Themes></FeatureLayer></Profile>"
  }
]
}

```

```

    }'''
113
def update_query(record_id, distance, time, emissions, cenEmissions,
114
toll_distance, toll_cost, manoeuvres):
    return '''UPDATE [dbo].[mn_trips]
115
SET [processed] = 1
116
    ,[distance_m] = '''+str(distance)+'''
117
    ,[time_s] = '''+str(time)+'''
118
    ,[hydrocarbons_g] = '''+str(emissions['hydrocarbons'])+''''
119
    ,[methane_g] = '''+str(emissions['methane'])+''''
120
    ,[hydrocarbonsExMethane_g] = '''+str(emissions['hydrocarbonsExMethane
    '])+''''
121
    ,[carbonMonoxide_g] = '''+str(emissions['carbonMonoxide'])+''''
122
    ,[carbonDioxide_g] = '''+str(emissions['carbonDioxide'])+''''
123
    ,[sulphurDioxide_g] = '''+str(emissions['sulphurDioxide'])+''''
124
    ,[nitrogenOxides_g] = '''+str(emissions['nitrogenOxides'])+''''
125
    ,[nitrousOxide_g] = '''+str(emissions['nitrousOxide'])+''''
126
    ,[ammonia_g] = '''+str(emissions['ammonia'])+''''
127
    ,[benzene_g] = '''+str(emissions['benzene'])+''''
128
    ,[toluene_g] = '''+str(emissions['toluene'])+''''
129
    ,[xylene_g] = '''+str(emissions['xylene'])+''''
130
    ,[lead_g] = '''+str(emissions['lead'])+''''
131
    ,[particles_g] = '''+str(emissions['particles'])+''''
132
    ,[fuel_kg] = '''+str(emissions['fuel'])+''''
133
    ,[cenEmissions_energyUseTank2Wheel_MJ] = '''+str(cenEmissions['
    energyUseTank2Wheel'])+''''
134
    ,[cenEmissions_energyUseWell2Wheel_MJ] = '''+str(cenEmissions['
    energyUseWell2Wheel'])+''''
135
    ,[cenEmissions_energyUseWell2Tank_MJ] = '''+str(cenEmissions['
    energyUseWell2Tank'])+''''
136
    ,[cenEmissions_co2eTank2Wheel_kg] = '''+str(cenEmissions['
    co2eTank2Wheel'])+''''
137
    ,[cenEmissions_co2eWell2Wheel_kg] = '''+str(cenEmissions['
    co2eWell2Wheel'])+''''
138
    ,[cenEmissions_co2eWell2Tank_kg] = '''+str(cenEmissions['
    co2eWell2Tank'])+''''
139
    ,[toll_distance_m] = '''+str(toll_distance)+'''
140
    ,[toll_cost_euro] = '''+str(toll_cost)+'''
141
    ,[manoeuvres] = '''+str(manoeuvres)+'''
142
WHERE Id='''+str(record_id)
143
DF = pd.read_sql_query(
144
'''SELECT [Id]
145
    ,[from_centralized_lat]
146
    ,[from_centralized_lng]
147
    ,[to_centralized_lat]
148
    ,[to_centralized_lng]
149
FROM [RPA_DEV].[dbo].[mn_trips]
150
WHERE PROCESSED IS NULL OR manoeuvres IS NULL
151
ORDER BY 1 ''', conn)
152
153
request_url = 'http://ptv.raben-group.com:50030/xroute/rs/XRoute/
154
calculateAdvancedTour'
emissions = ['hydrocarbons', 'methane', 'hydrocarbonsExMethane', '
155
carbonMonoxide', 'carbonDioxide', 'sulphurDioxide', 'nitrogenOxides', '
nitrousOxide', 'ammonia', 'benzene', 'toluene', 'xylene', 'lead', 'particles',
'fuel', 'nitrogenDioxide', 'fossilCarbonDioxide', 'particleNumber']
cenEmissions = ['energyUseTank2Wheel', 'energyUseWell2Wheel', '
156
energyUseWell2Tank', 'co2eTank2Wheel', 'co2eWell2Wheel', 'co2eWell2Tank'
]
157
for index, row in tqdm(DF.iterrows(), total=DF.shape[0]):
158
    dict_emissions = {};
159
    dict_cenEmissions = {};
160
    toll_distance = 0;
161
    toll_cost = 0;
162
    manoeuvres = 0;
163
    manoeuvres = 0;
164
    try:
165
        response = requests.post(request_url, json = json.loads(
166
167

```

```

        http_request_string(row['from_centralized_lng'],
                             row['from_centralized_lat'],
                             row['to_centralized_lng'],
                             row['to_centralized_lat']))
    if response is not None:
        response_json = response.json()

        distance = response_json['route']['info']['distance']
        time = response_json['route']['info']['time']
        manoeuvres = len(response_json['route']['manoeuvres'])

        for e in emissions:
            dict_emissions[e] = response_json['route']['emissions'][e]
        for c in cenEmissions:
            dict_cenEmissions[c] = response_json['route']['cenEmissions']
            ][ 'basedOnHBEFA' ][c]
        for d in response_json['countryInfos'][0][ 'perTypeTollDistance'
            ]:
            toll_distance += int(d)
        for p in response_json['countryInfos'][0][ 'perTypeTollPrice' ]:
            toll_cost += int(p)

        cursor.execute(update_query(row['Id'], distance, time,
                                     dict_emissions, dict_cenEmissions, toll_distance, toll_cost,
                                     manoeuvres))
        conn.commit()
    except Exception as e:
        print(str(e))
DF_Result = pd.read_sql_query(
    '''SELECT *
    FROM [RPA_DEV].[dbo].[mn_trips]
    ORDER BY 1 ''', conn)
DF_Result.to_excel('praca_magisterska_trasy_final.xlsx')
conn.close()

#LOAD NEEDED PACKAGES
import statsmodels.api as sm
import pandas as pd
import numpy as np
import gme as gme
import matplotlib.pyplot as plt
import geopandas as gpd
import folium
from mpl_toolkits.axes_grid1 import make_axes_locatable
import ggravity as ge

#SHOW ALL COLUMNS
pd.set_option('display.max_colwidth', None)
#CHANGE FORMAT OF FLOATS
pd.set_option('display.float_format', '{:20,.10f}'.format)

#LOAD SHP MAPS
mapa_woj = gpd.read_file(r'SHP\Wojewodztwa.shp')
mapa_woj = mapa_woj[['JPT_NAZWA_', 'geometry']]

#LOAD DATA
df_init = pd.read_csv('https://media.githubusercontent.com/media/
mikolajnowak97/MasterThesis_The-use-of-the-gravity-model-in-logistics/
main/Data/04_trips_calculated.csv', sep=';')

#TRANSFORM COLUMNS
df_init['year'] = 2021
df_init['distance_km'] = np.where(df_init['distance_m']>0, df_init['
distance_m']/1000, 0)
df_init['toll_distance_km'] = np.where(df_init['toll_distance_m']>0,

```

```

df_init['toll_distance_m']/1000, 0)
df_init['toll_cost_euro'] = np.where(df_init['toll_cost_euro']>0, df_init[229
    'toll_cost_euro']/100, 0)
df_init['time_h'] = np.where(df_init['time_s']>0, df_init['time_s']/3600, 230
    0)

#CHOOSE COLUMNS TO MODEL 231
df = df_init[['from_province', 'to_province', 'distance_m', 'distance_km', 232
    'toll_distance_m', 'toll_distance_km',
    'toll_cost_euro', 'time_s', 'time_h', 'manoeuvres', ' 234
    hydrocarbons_g',
    'methane_g', 'hydrocarbonsExMethane_g', 'carbonMonoxide_g', ' 235
    carbonDioxide_g', 'sulphurDioxide_g',
    'nitrogenOxides_g', 'nitrousOxide_g', 'ammonia_g', 'benzene_g' 236
    ', 'particles_g', 'fuel_kg', 'year']]

#DEFINE MODEL VARIABLES 237
variables = ['hydrocarbons_g', 'methane_g', 'carbonMonoxide_g', ' 238
    carbonDioxide_g', 'sulphurDioxide_g',
    'nitrogenOxides_g', 'nitrousOxide_g', 'ammonia_g', 'benzene_g', 239
    'particles_g', 'fuel_kg']

#GRAVITY MODEL FOR EACH VARIABLE 240
for variable in variables: 241
    #Define a gme EstimationData object 242
    gme_data = gme.EstimationData(data_frame = df, 243
        imp_var_name = 'from_province', 244
        exp_var_name = 'to_province', 245
        trade_var_name = variable, 246
        year_var_name = 'year') 247

    #Create and estimate a gme EstimationModel gravity model 248
    gme_model = gme.EstimationModel(estimation_data = gme_data, 249
        lhs_var = variable, 250
        rhs_var = ['distance_km', 251
            'time_h', 252
            'toll_distance_km', 253
            'toll_cost_euro', 254
            'manoeuvres'], 255
        fixed_effects = ['from_province', ' 256
            to_province']) 257

    estimates = gme_model.estimate() 258
    results = estimates['all'] 259

    df_coef = results.params.to_frame() 260

    #Save table of results summary 261
    print(results.summary()) 262
    with open('outputs/'+variable+'_summary.txt', 'w', encoding='utf-8') as 263
        f: 264
        f.write(results.summary().as_latex()) 265

#RESULTS VISUALIZATION 266
#Prepare data to plot 267
df_coef.reset_index(level=0, inplace=True) 268
df_coef = df_coef.rename(columns={'index': 'woj'}) 269
df_coef = df_coef[df_coef['woj'].str.contains('from_province_fe_', 270
    regex= True, na=False)].replace(to_replace = 'from_province_fe_', 271
    value = '', regex = True) 272
data_map = pd.merge(mapa_woj, df_coef, how='left', left_on='JPT_NAZWA_', 273
    , right_on='woj') 274
data_map = data_map.to_crs(epsg=2180) 275

#Create plot 276
fix, ax = plt.subplots(1, figsize=(8,8)) 277
divider = make_axes_locatable(ax) 278
cax = divider.append_axes("right", size="5%", pad=0.1) 279
data_map.plot(column=0, 280

```

```
ax=ax, 285
cax=cax, 286
cmap='YlOrRd', 287
linewidth=0.8, 288
edgecolor='gray', 289
legend=True) 290
ax.axis('off') 291
plt.savefig('outputs/'+variable+'_coef.png') 292
```
