

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

FAKULTA ŽIVOTNÍHO PROSTŘEDÍ

KATEDRA PROSTOROVÝCH VĚD



POVRCHOVÉ TEPELNÉ OSTROVY EVROPSKÝCH
METROPOLÍ A ANALÝZA JEJICH ŘÍDÍCÍCH FAKTORŮ

BAKALÁŘSKÁ PRÁCE

Vedoucí práce: Ing. David Moravec, Ph.D.

Bakalant: Kryštof Vávra

2022

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Fakulta životního prostředí

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Kryštof Vávra

Geografické informační systémy a dálkový průzkum Země v životním prostředí

Název práce

Povrchové tepelné ostrovy evropských metropolí a analýza jejich řídicích faktorů

Název anglicky

Surface urban heat islands in European capitals and analysis of their driving factors

Cíle práce

Hlavním cílem bakalářské práce je analyzovat efekt povrchových tepelných ostrovů v evropských metropolích a klasifikovat je dle odolnosti na základě volně dostupných dat ze senzoru MODIS. Dále se bakalářská práce věnuje problematice řídicích faktorů tepelných ostrovů. Pro zpracování této části je potřeba vybrat relevantní řídicí faktory, analyzovat jejich efekt na jednotlivá města a analyzovat vztahy mezi faktory a intenzitou povrchového tepelného ostrova měst.

Metodika

Pro splnění stanovených cílů bakalářské práce bude využita platforma Google Earth Engine (GEE) a program IBM SPSS. Na platformě GEE bude pomocí jazyka JavaScript vypracováno několik skriptů. Každý skript bude řešit specifickou úlohu a vytvoří potřebné grafické a datové výstupy, které budou následně vhodně interpretovány, nebo dále zpracovány v programu SPSS.

Rámcový postup:

- 1) Import dat
- 2) Filtr dat
- 3) Úprava dat na správné hodnoty
- 4) Identifikace městských a venkovských oblastí
- 5) Výpočet intenzity povrchového tepelného ostrova jednotlivých měst
- 6) Vytvoření grafů nebo vzorků
- 7) Export grafů nebo vzorků

Doporučený rozsah práce

30 stran

Klíčová slova

povrchové tepelné ostrovy měst, řídicí faktory, dálkový průzkum země, MODIS, evropské metropole

Doporučené zdroje informací

- Dewan, Ashraf, et al. "Surface urban heat island intensity in five major cities of Bangladesh: Patterns, drivers and trends." *Sustainable Cities and Society* 71 (2021): 102926.
- Peng, S., Piao, S., Ciais, P., Friedlingstein, P., Oettle, C., Bréon, F.-M., ... Myneni, R. B. (2011). Surface Urban Heat Island Across 419 Global Big Cities. *Environmental Science & Technology*, 46(2), 696–703. doi:10.1021/es2030438
- Zhou, Decheng, et al. "Surface urban heat island in China's 32 major cities: Spatial patterns and drivers." *Remote sensing of environment* 152 (2014): 51-61.

Předběžný termín obhajoby

2021/22 LS – FZP

Vedoucí práce

Ing. David Moravec, Ph.D.

Garantující pracoviště

Katedra prostorových věd

Elektronicky schváleno dne 22. 3. 2022

doc. Ing. Petra Šímová, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 22. 3. 2022

prof. RNDr. Vladimír Bejček, CSc.

Děkan

V Praze dne 30. 03. 2022

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci na téma: Povrchové tepelné ostrovy evropských metropolí a analýza jejich řídicích faktorů vypracoval samostatně a citoval jsem všechny informační zdroje, které jsem v práci použil a které jsem rovněž uvedl na konci práce v seznamu použitých informačních zdrojů.

Jsem si vědom, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, ve znění pozdějších předpisů, především ustanovení § 35 odst. 3 tohoto zákona, tj. o užití tohoto díla.

Jsem si vědom, že odevzdáním bakalářské práce souhlasím s jejím zveřejněním podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů, ve znění pozdějších předpisů, a to i bez ohledu na výsledek její obhajoby.

Svým podpisem rovněž prohlašuji, že elektronická verze práce je totožná s verzí tištěnou a že s údaji uvedenými v práci bylo nakládáno v souvislosti s GDPR.

V Praze dne 20.3.2020

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucímu práce, Ing. Davidovi Moravcovi, Ph.D. za věcné připomínky a cenné rady při zpracování této bakalářské práce. Také bych chtěl poděkovat své rodině a zejména své přítelkyni Cloé Moreno za trpělivost a podporu.

ABSTRAKT

Bakalářská práce se zabývá problematikou městských tepelných ostrovů (UHI), které jsou jedny z nejevidentnějších projevů lidské činnosti na Zemi. Práce zkoumá noční a denní intenzity povrchových tepelných ostrovů (SUHII) v 10 evropských metropolích pomocí satelitních dat MODIS z let 2015 až 2020. Dále se práce věnuje potenciálním řídicím faktorům, které na povrchové tepelné ostrovy měst (SUHI) působí a vyhodnocuje jejich vliv.

Na platformě Google Earth Engine (GEE) byly na základě dat krajinného pokryvu ve všech zkoumaných městech identifikovány městské a venkovské oblasti. Teploty těchto oblastí od sebe byly odečteny a tím byly získány snímky reprezentující intenzity SUHI v jednotlivých městech. Pro všechny městské oblasti byly dále vypracovány snímky s hodnotami řídicích faktorů, ze kterých byly společně se snímky SUHII vytvořeny vzorky. Nakonec byla na základě těchto vzorků provedena vícenásobná lineární regrese.

Práce odhalila, že všechny zkoumaná města jsou vystavena efektu povrchových tepelných ostrovů. Jako nejvíce zasažená města byly identifikovány Londýn a Paříž. Velmi odlišně se projevoval Madrid, který měl negativní intenzity SUHI během dne, a naopak velmi vysoké intenzity během noci. Ze studie vyplývá, že města by se měla především zaměřit na správný výběr materiálu při výstavbě a na zvýšení vegetačního pokryvu. Tyto dva faktory byly v práci identifikovány jako klíčové pro zmírnění negativních dopadů UHI.

KLÍČOVÁ SLOVA:

SUHI, dálkový průzkum Země, Google Earth Engine, MODIS, vícenásobná lineární regrese

ABSTRACT

The bachelor thesis deals the issue of urban heat island (UHI), which is one of the most obvious manifestations of human activity on Earth. The work examines the nighttime and daytime intensities of surface urban heat islands (SUHI) in 10 European capitals using MODIS satellite data from 2015 to 2020. Furthermore, the thesis also deals with potential driving factors that affect the surface urban heat islands (SUHI) and evaluates their impact on the intensity of SUHI.

On the Google Earth Engine (GEE) platform, first urban and rural areas were identified based on land cover data in all surveyed cities. The temperatures of these areas were subtracted from each other to obtain images representing the intensities of SUHI in each city. Images of driving factors values were further created for all urban areas, from which samples were created together with SUHI images. Finally, multiple linear regression was performed on these samples.

The work revealed that all the cities are exposed to the effect of surface urban heat islands. London and Paris were identified as the most affected cities. Results in Madrid were very different than in other cities, with intensities of SUHI being negative during the day and positive during the night. The study shows that cities should focus on choosing the right materials for construction and increase the vegetation coverage around the city. These two factors were identified in the work as the key for mitigating the negative impacts of UHI.

KEYWORDS:

SUHI, remote sensing, Google Earth Engine, MODIS, multiple linear regression

Obsah

1. Úvod	10
2. Cíle práce	10
3. Literární rešerše	11
3.1 Vznik městských tepelných ostrovů	11
3.2 Typy městských tepelných ostrovů a techniky jejich pozorování	12
3.3 Negativní dopady městských tepelných ostrovů	14
3.4 Adaptace měst na efekt městských tepelných ostrovů	14
3.5 SUHI a jeho řídicí faktory	15
3.5.1 Vegetace	15
3.5.2 Nadmořská výška	16
3.5.3 Populace	16
3.5.4 Voda	16
3.5.5 Vyzařování světla	17
3.5.6 Albedo	17
3.5.7 Nepropustné povrchy	17
3.6 Dostupná data pro SUHI	18
3.7 Regresní analýza	18
3.7.1 Metoda Stepwise	18
4. Metodika	18
4.1 Mechanismy skriptů	19
4.2 Zájmové území	19
4.3 Identifikace městských a venkovských oblastí	20
4.3 Výpočet povrchové teploty a intenzity povrchových tepelných ostrovů měst ..	21
4.4 Výpočet řídicích faktorů	22
4.5 Vytvoření grafických a datových výstupů	24
4.6 Regresní analýza	25
5. Výsledky	26
5.1 Intenzity povrchových tepelných ostrovů	26

5.1.1 Noční intenzita povrchového tepelného ostrova.....	26
5.1.2 Denní intenzita povrchového tepelného ostrova.....	27
5.2. Analýza řídicích faktorů SUHI.....	29
5.2.1 Průměrné hodnoty řídicích faktorů v městských oblastech.....	29
5.2.2 Výsledky vícenásobné lineární regrese	29
6. Diskuse.....	33
6.1 Zhodnocení metodiky	33
6.2. Zhodnocení a porovnání výsledků	34
6.2.1. Intenzita SUHI ve městech.....	34
6.2.2. Vícenásobná lineární regrese	34
7. Závěr a přínos práce	35
8. Přehled literatury a použitých zdrojů	36
10. Seznam příloh	39

1. Úvod

Poslední dobou zažíváme na naší planetě velké klimatické změny. Vědci odhadují, že teplota Země dále poroste kvůli skleníkovým plynům vyprodukovaným lidskou činností (Khare et al, 2021). Města a jejich obyvatelé se díky rozsáhle expanzi měst v poslední době stávají klíčovými hybateli globálních změn klimatu (Dewan et al., 2021). Urbanizace je jedním z nejvíce evidentních vlivů lidí na Zemi. Města, a především jejich centra, dosahují vyšších teplot než okolní venkovské oblasti, tento fenomén je znám jako efekt městského tepelného ostrova (UHI) (Peng et al., 2011).

UHI má širokou škálu negativních dopadů. Ovlivňuje například biologickou rozmanitost, kvalitu vody, znečištění vzduchu, a nepochybně i zdraví a komfort obyvatel (Zhou et al., 2014). Z toho důvodu je pro lidstvo důležité porozumět tomu, které faktory jsou pro UHI klíčové a mají na UHI největší vliv. Tyto znalosti pak můžou být v budoucnu užitečné pro správný rozvoj měst a snížení negativních dopadů UHI (Wang et al., 2021).

V praxi jsou rozlišovány dva hlavní typy tepelných ostrovů: atmosférický tepelný ostrov města (AUHI) a povrchový tepelný ostrov města (SUHI). Tyto typy se od sebe odlišují především výškou nad zemí, ve které jsou pozorovány a měřeny (Dewan et al., 2021). Dálkový průzkum Země umožňuje pozorovat SUHI pomocí měření teplot povrchu (LST). SUHI je pak definováno jako rozdíl teplot povrchu mezi městskými oblastmi a venkovskými oblastmi (Peng et al., 2011).

2. Cíle práce

Hlavním cílem bakalářské práce je analyzovat efekt povrchových tepelných ostrovů v evropských metropolích, zkoumat jejich časový trend a klasifikovat je dle odolnosti na základě volně dostupných dat ze senzoru MODIS.

Dále se bakalářská práce věnuje problematice řídicích faktorů tepelných ostrovů. Pro zpracování této části je potřeba vybrat relevantní řídicí faktory, nalézt pro ně vhodná data a analyzovat jejich efekt na jednotlivá města.

Na základě těchto výsledků se práce pokusí zjistit hlavní příčinu povrchových tepelných ostrovů a definovat efektivní strategii ke zmírnění negativních dopadů.

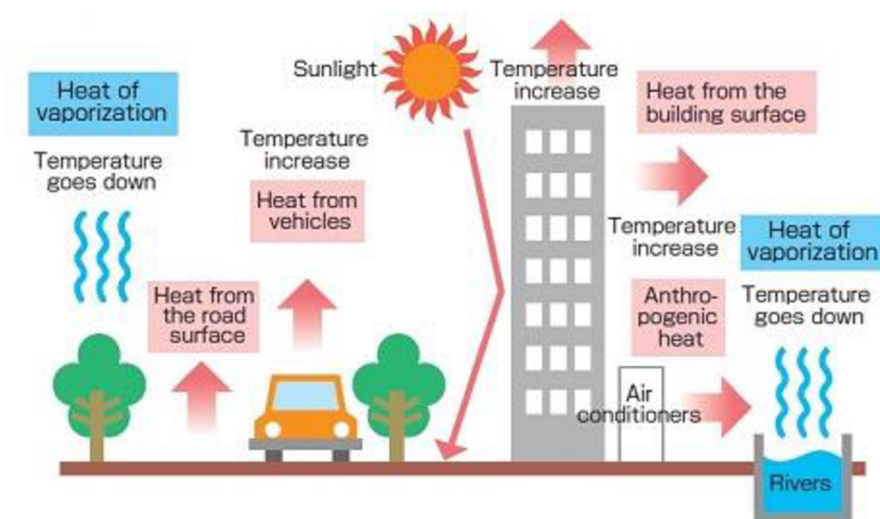
3. Literární rešerše

3.1 Vznik městských tepelných ostrovů

Po většinu lidské historie žili lidé po celém světě v malých komunitách. To se ale během posledního století dramaticky změnilo, kvůli obrovské migraci lidí z venkovských do městských oblastí. V současnosti žije více jak 56 % populace v městských oblastech, a to má za následek intenzivní rozšiřování měst a zhušťování již obydlených částí měst (Ritchie & Roser, 2018). Velká část městské populace nyní žije v oblastech, kde se pravidelně vystavuje velmi vysokým teplotám, což může mít velký vliv na zdraví obyvatel (Marando et al., 2022).

Postupnou urbanizací jsou přírodní povrchy nahrazovány umělými materiály jako jsou cement, asphalt nebo beton. Tyto povrchy více absorbují dopadající sluneční záření, akumuluji teplo, omezují proudění vzduchu a zamezují vsakování vody. Teplo se pak postupně uvolňuje do atmosféry a dochází k intenzivnímu ohřívání vzduchu ve městě (Wang et al, 2021). Teplotu ve městě navíc kromě jiných faktorů významně ovlivňuje i geografická poloha města, jeho rozloha a geometrie, počet obyvatel anebo antropogenní vlivy jako například vozidla, klimatizace, nebo domácí topení, které uvolňují své teplo do atmosféry.

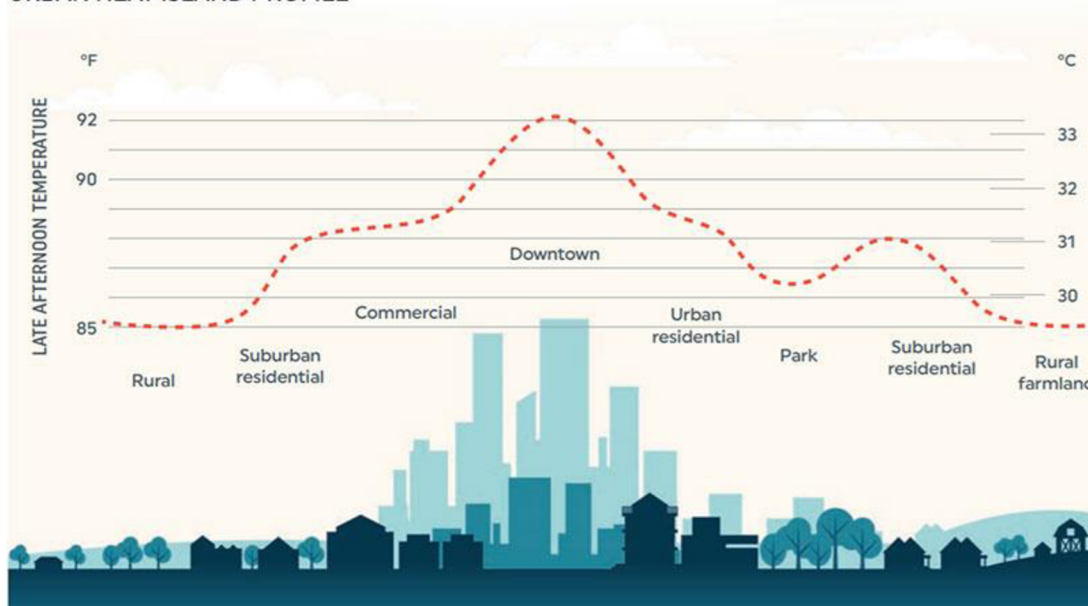
Okolní venkovské oblasti jsou narušeny od těch městských tvořené převážně přirozenými povrchy, ty mají obecně vyšší odrazivost, a proto se tolik neohřívají. Navíc mají větší poměr vegetace, která díky procesu evapotranspirace akumuluje méně tepla a ochlazuje své okolní prostředí (Vacek et al, 2018).



Obr. č. 1: Vznik městských tepelných ostrovů (Zdroj: Casandra, 2013)

Města vytváří v krajině “ostrov”, které mají odlišnou teplotu oproti okolním venkovským oblastem, tento rozdíl v teplotách se nazývá městský tepelný ostrov a je považován za jeden z nejviditelnějších charakteristik městského klimatu. Čím větší je rozdíl teplot, tím je větší i intenzita tepelného ostrova. Ve speciálních případech může docházet k nižším teplotám ve městě oproti venkovským oblastem, tento jev se pak nazývá městský studený ostrov (UCI). Efekt městských tepelných ostrovů je různý v závislosti na ročním období a denním času, kdy je zkoumán (Memon et al, 2007; Zhang & Wang, 2008; Khare et al, 2021).

URBAN HEAT ISLAND PROFILE



Obr. č. 2: Vývoj teploty v městských a venkovských oblastech (Zdroj: Foster, 2020)

3.2 Typy městských tepelných ostrovů a techniky jejich pozorování

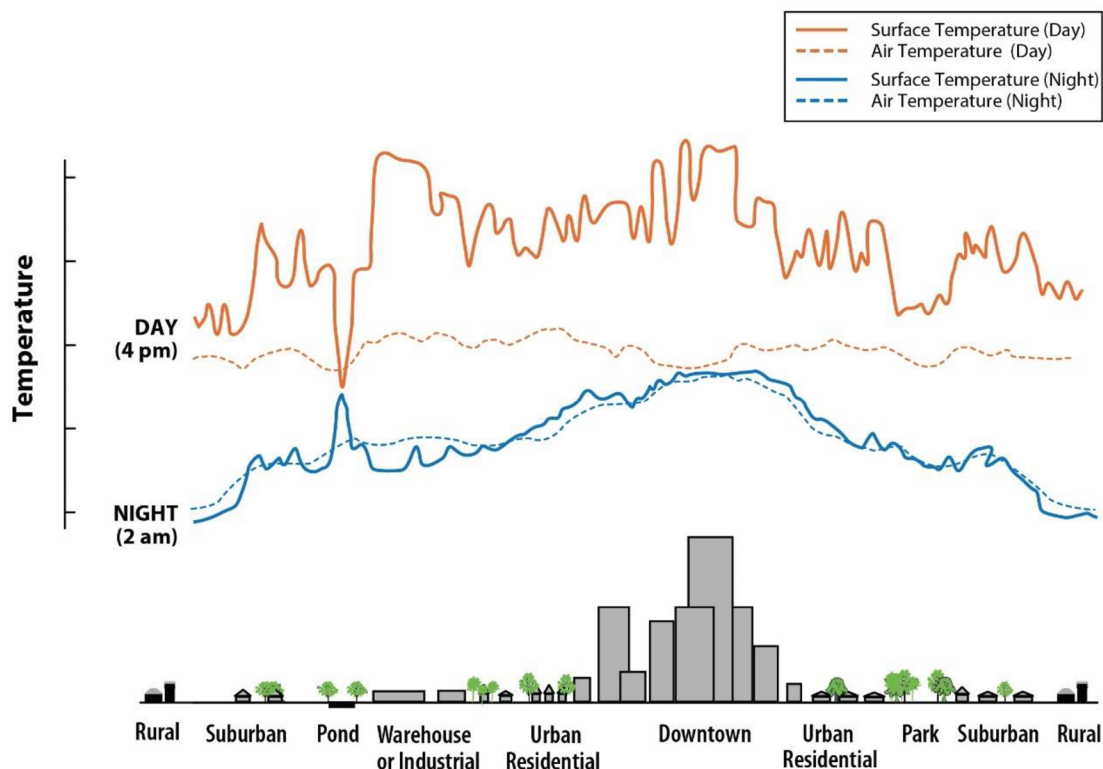
Teploty jsou většinou různé na povrchu a ve vzduchu nad povrchem. Existují tedy dva typy tepelných ostrovů, které se od sebe liší především vzdáleností od země, ve které byly naměřeny a technikou, která je používána pro jejich měření a identifikaci (EPA, 2021).

Atmosférický tepelný ostrov města (AUHI) odkazuje na rozdílné teploty vzduchu v městských a venkovských oblastech. Rozděluje se ve vertikálním směru do dvou kategorií podle vzdálenosti od povrchu, a to na tepelný ostrov v přízemní vrstvě atmosféry, který dosahuje do průměrné výšky střech budov a tepelný ostrov v mezní vrstvě atmosféry, který se nachází nad přízemní vrstvou. Tepelné ostrovy v atmosféře se pozorují například pomocí pozemních měření teplot vzduchu na meteorologických stanicích ve městě a venkovských oblastech a poté se teploty

porovnávají. (Stathopoulou & Cartalis, 2007; OKE, 1982). Během noci se efekty UHI stávají evidentnějšími díky větší tepelné setrvačnosti materiálů používaných v městské zástavbě. Právě v noci může teplo akumulované v městských materiálech, zejména v souvislosti s vlnami veder, vytvářet významná zdravotní rizika (Arellano & Roca, 2021).

Povrchový tepelný ostrov města je rozdíl mezi teplotou povrchu v městské a venkovské oblasti. Pozorování se provádí převážně pomocí dat z dálkových měření s využitím tepelných senzorů, které nám umožňují získávat data o teplotě svrchní vrstvy povrchu. Hlavní výhody dálkových pozorování je široké prostorové pokrytí a schopnost snímání stejné oblasti i několikrát denně. Často se k těmto pozorování využívají satelitní data ze senzorů MODIS, Landsat nebo NOAA (Arrau & Peña, 2021; Stathopoulou & Cartalis, 2007).

Teploty povrchu jsou běžně vyšší, než jakých dosahuje vzduch nad povrchem a jsou značně ovlivněny strukturou a barvou daného povrchu. Rozdíly teplot jsou výraznější během dne, a to především v letních měsících (Stathopoulou & Cartalis, 2007). To se ale neprokázalo ve studii Zhou et al. (2014), ve které se zaměřili na zkoumání efektu SUHI ve velkých městech Číny. V této studii roční intenzity povrchových tepelných ostrovů měst (SUHI) dosahovaly od 0,01 °C do 1,87 °C pro denní pozorování a od 0,35 °C do 1,95 °C pro noční pozorování. Největší SUHI dosahovaly jihovýchodní a severní regiony s tím, že se intenzita velmi lišila v závislosti na ročním období. Při denních pozorování byla zaznamenána větší intenzita v létě a při nočních pozorování v zimě. Oproti tomu Dewan et al. (2021) zkoumali SUHI v 5 bangladéšských městech a zde byl roční průměr SUHI nejvyšší pro denní pozorování (2,74 °C) s tím, že nejvyšších hodnot dosahovaly centra měst. Peng et al. (2011) se zaměřili na výzkum SUHI ve 419 světových městech a zjistili, že průměrná roční intenzita pro denní pozorování byla v 74 % městech větší než pro noční pozorování. Intenzita SUHI byla ve studii v 92 % městech pozitivní s tím, že města s negativní denní intenzitou se nacházela v pouštních oblastech.



Obr. č. 3: Vývoj teploty povrchu a vzduchu ve dne a v noci (Zdroj: Earth Resources Observation, 2019)

3.3 Negativní dopady městských tepelných ostrovů

Zvýšené letní teploty ve městě zhoršují schopnost lidí vyrovnávat teplo pomocí systému termoregulace. To vede ke zvýšení zdravotních rizik jako je vyčerpání, úpal a v nejhorším případě až smrt. Kromě toho mohou zvýšené teploty zhoršit zdravotní stav u lidí s chronickými zdravotními problémy, například pokud má dotčený člověk problémy se srdcem nebo plícemi. V největším ohrožení jsou tak zejména starší lidé, kteří na tyto problémy trpí (Buchin et al, 2016).

Tepelné ostrovy mají vliv nejen na zdraví obyvatel, ale i na jejich pracovní efektivitu a pozornost. Tyto vlastnosti se často zhoršují díky zvýšené únavě, která se může dostavit při extrémních teplotách. Navíc mohou ovlivnit i ekonomickou výkonnost a celkový komfort a kvalitu života obyvatel (Vacek et al, 2018).

UHI navíc v létě zvyšují nároky na chladicí prostředky, což vede k větší spotřebě elektřiny a větší produkci skleníkových plynů a znečištění, a to významně přispívá ke zhoršení ovzduší (Khare et al, 2021).

3.4 Adaptace měst na efekt městských tepelných ostrovů

Zmírnění efektu tepelného ostrova může přinést spoustu výhod na lokální i globální úrovni. Při správné adaptaci měst se může například zlepšit kvalita vzduchu

a vody nebo zmírnit spotřeba elektřiny v letních měsících a snížit potenciální zdravotní rizika.

Existuje mnoho strategií pro zmírnění efektu UHI. Jedna z nich uvádí využití takzvaných studených střech, které mají schopnost efektivně odrážet sluneční záření a vyzařovat teplo v určitém spektru a tím snížit absorpci slunečního záření. Další strategie navrhuje výstavbu zelených střech a zdí, ty jsou porostlé vrstvou vegetace, která díky svým specifickým vlastnostem zmírňuje teploty na povrchu. Více strategií se pak zaměřuje na vysazování stromů, především na otevřených prostranstvích, nebo na vytváření vodních ploch, které by ochlazovaly vzduch pomocí procesu evapotranspirace. Existují také strategie pro výstavbu speciální dlažby, která pomáhá snižovat teplotu ve městě (Khare et al., 2021).

3.5 SUHI a jeho řídicí faktory

Porozumění řídicích faktorů, které ovlivňují intenzitu městských ostrovů je klíčové k vytvoření adaptací a plánů, které mohou vést ke snížení negativních dopadů tepelného ostrova (Wang et al, 2021).

3.5.1 Vegetace

Vegetace má schopnost ochlazovat své okolní prostředí díky procesu evapotranspirace z jejich listů, vyššímu albedu, nižší tepelné kapacity a schopnosti stínění povrchu (Tan et al., 2021). Často se k identifikaci oblastí pokryté vegetací a zdraví vegetace používá **Normalizovaný Diferenční Vegetační Index (NDVI)**, Tento populární index popisuje rozdíl mezi odraženým červeným a blízkým infračerveným světlem vegetace. Nabývá hodnot od (-1) do (+1), ty udávají množství chlorofylu, který se nachází ve vegetaci. Vyšší hodnoty indikují hustší a zdravější vegetaci (Weier & Herring, 2000; Rani et al., 2018). Hodnoty NDVI se mění v závislosti na ročním období. Největších hodnot nabývá v létě, díky bujné vegetaci a nejmenších v zimě, když je chladno a většina vegetace je suchá (Hu et al., 2020). Další často používaný index je **Vylepšený Vegetační Index (EVI)**. Jde o upravenou verzi NDVI, která se vyznačuje vyšší citlivostí na biomasu a půdní podmínky. EVI používá modré, červené a blízké infračervené pásmo a nabývá hodnot od (-1) do (+1) (Basu & Das, 2021).

Zhou et al. (2014) v jejich práci potvrdili, že intenzita denního SUHI v letním období úzce souvisí s hodnotami EVI. K podobným výsledkům došli i ve studii Peng et al. (2011), kde zjistili, že NDVI významně negativně koreluje s ročními průměry denních pozorování SUHI, ale při porovnávání NDVI s nočními pozorováními SUHI

nenášli nijak významnou korelaci. Oproti tomu v práci Dewan et al. (2021) našli negativní vztah EVI s denními i nočními SUHII.

3.5.2 Nadmořská výška

Obecně má teplota tendenci klesat s rostoucí nadmořskou výškou (Peng et al., 2020), to ale neplatí o teplotě povrchu, ta naopak stoupá s nadmořskou výškou, a to především během dne. (Lome et al., 2019). Pro práci s nadmořskými výškami se v praxi často využívá **Digitální Model Terénu (DMT)**. DMT je digitální model reálného povrchu, který reprezentuje nadmořské výšky (Wang et al., 2018). Data pro DMT mohou být získány různými způsoby, například pomocí fotogrammetrie, LiDARu, dálkovým snímáním, nebo pozemním měřením (Gandhi & Sarkar, 2016).

Zhou et al. (2014) zkoumali vliv nadmořských výšek na SUHII a neodhalili žádný významný vztah SUHII s nadmořskými výškami ve městech, kde je relativně malá změna v nadmořských výškách, to stejné tvrdí i Wang et al. (2021).

3.5.3 Populace

Lidé mají hlavní roli při utváření měst, vyšší počet obyvatel v městských oblastech má vliv na velikost města a přeměnu přírodních povrchů na povrchy umělé. Počet obyvatel nepřímo ovlivňuje intenzitu vývoje území a komplexnost městského povrchu. Větší počet lidí znamená i nárůst teploty z lidských organismů a zvýšení spotřeby energie, která se pak uvolňuje ve formě tepla do atmosféry (Buchin et al., 2016; Dewan et al., 2021).

Peng et al. (2011) ani Wang et al. (2021) nenašli žádný významný vztah mezi hustotou obyvatel a SUHII. Na rozdíl od předchozích studií Dewan et al. (2021) našli významný vztah hustoty obyvatel s denní SUHII.

3.5.4 Voda

Voda se vyznačuje vyšší tepelnou kapacitou než ostatní povrchy. Dokáže lépe absorbovat sluneční záření během dne a pomáhá ochlazovat své okolí. Během noci ale vodní plochy uvolňují teplo do atmosféry, to naopak přispívá k vyšším teplotám (Mathew et al., 2022). Často je při zkoumání efektu vody používán **Normalizovaný Diferenční Vodní Index (NDWI)**. To je index, který úzce souvisí s obsahem vody v rostlinách. Je odvozený ze satelitních snímků blízkého infračerveného (NIR) pásma a krátkého infračerveného (SWIR) pásma. Nabývá hodnot od (-1) do (+1), které udávají obsah vody v listech na úrovni koruny stromů (Gao, 1996).

Podle studie Dewan et al. (2021) existuje u NDWI významný pozitivní vztah s denní intenzitou SUHI. Naproti tomu výsledky Xu et al. (2013) ukazují negativní vztah NDWI se SUHI.

3.5.5 Vyzařování světla

Vyšší intenzita vyzařovaného světla se běžně projevuje vyšší teplotou v okolní oblasti a indikuje zvýšenou lidskou činnost. Tento jev se běžně více projevuje během noci než během dne (Li et al., 2019).

Peng et al. (2011) našli pozitivní korelaci SUHI s nočním zářením světla. Ve studii Zhou et al. (2014) vysvětlovalo v letních měsících noční záření světla denní SUHI nejlépe ze všech faktorů.

3.5.6 Albedo

Albedo je míra odrazivosti. Jde o část dopadajícího záření, které je povrchem odraženo. Čím větší je hodnota albeda, tím větší je odrazivost daného povrchu. Povrchy mají různé hodnoty albeda, tmavé městské povrchy mají nižší hodnoty albeda než přírodní povrchy. To zvyšuje absorpci slunečního záření, což přispívá ke zvýšení teploty povrchu a vzduchu. Venkovské oblasti jsou tvořené především přírodními povrchy, které mají vyšší albedo než městské povrchy. To znamená, že naopak lépe odráží sluneční záření a snižuje absorpci (Trlica et al., 2017).

Ve studii Peng et al. (2011) platilo, že průměrná roční intenzita nočních pozorování SUHI měla významný pozitivní vztah s albedem. Ve výzkumu Dewan et al. (2021) se ukázalo, že byl sice vztah SUHI a albeda nekonzistentní, ale i zde existoval signifikantní pozitivní vztah s denním pozorování SUHI. V práci Zhou et al. (2014) bylo albedo faktorem, který nejlépe vysvětloval noční intenzitu SUHI, a to nezávisle na ročním období.

3.5.7 Nepropustné povrchy

Nepropustné povrchy indikují nižší infiltraci a odpařování. Mají tendenci způsobovat vyšší teploty městského povrchu, který se skládá především z umělých povrchů jako je beton nebo asfalt (Wang et al., 2021).

Výsledky ze studie Dewan et al. (2021) ukazují signifikantní pozitivní vztah nepropustnosti a denní SUHI, a v práci Wang et al. (2021) dokonce identifikovali nepropustné povrchy jako hlavní příčinu SUHI.

3.6 Dostupná data pro SUHI

K dispozici je hned několik satelitů/senzorů, ze kterých je možné získat data o měření SUHI: Landsat, Sentinel 3, Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), MODerate resolution Imaging Spectroradiometer (MODIS), Advanced Very High Resolution Radiometer (AVHRR), National Oceanic and Atmospheric Administration (NOAA), ENVIronment SATellite (ENVISAT) (Bahi et al. 2020).

3.7 Regresní analýza

Statistické modely jsou často využívány k porozumění a předpovědi tepelných ostrovů. Vzhledem k tomu, že na tepelné ostrovy působí většinou hned několik faktorů najednou, je potřeba zkoumat efekt více nezávisle proměnných. Jeden z modelů, který toto umožňuje je **model vícenásobné lineární regrese**. Tento model umožňuje identifikovat významné faktory, které ovlivňují tepelný ostrov. Nevýhodou je, že předpokládá pouze lineární vztahy mezi proměnnými a je velmi citlivý na odlehle hodnoty (Oukawa et al., 2022).

3.7.1 Metoda Stepwise

Princip této metody spočívá v tom, že jsou do modelu postupně vkládány nezávisle proměnné. Nejdříve je do regresního modelu vložena proměnná s nejvyšší korelací vůči závislé proměnné. Poté je do nového modelu vložena předchozí proměnná společně s novou proměnnou, která má druhou nejvyšší korelací. Tento proces se opakuje, dokud nezůstává žádná signifikantní proměnná. Tato metoda umožňuje odstranit proměnnou, která byla zahrnuta v dřívějším kroku, ale již není významná vzhledem k nezávisle proměnným, které byly přidány později (Tranmer et al., 2020).

4. Metodika

Většina práce byla provedena na platformě Google Earth Engine (GEE), na které byly vypracovány tři skripty:

Skript (a) – Měsíční průměry SUHI ve městech (příloha č.1)

Skript (b) – Průměrné hodnoty řídicích faktorů a průměrné SUHI ve městech (příloha č.2)

Skript (c) – Vytvoření vzorků pro regresní analýzu (příloha č.3)

4.1 Mechanismy skriptů:

(podrobný popis skriptů je součástí skriptů pomocí komentářů v nich)

- 1) V horní části skriptu jsou definované proměnné, které slouží jako vstupní hodnoty do skriptu. Tyto proměnné umožňují uživateli ovládat skript. Uživatel má možnost určit, která města budou skriptem zpracována. Stejně tak může určit roky, měsíce a dny, ze kterých budou zpracovány snímky a rok ze kterého bude použit krajinný pokryv.
- 2) Skripty využívají mechanismus cyklů. Nejdříve skript vypracuje výsledky pro jednotlivé měsíce, výsledky z jednotlivých měsíců jsou ukládány do seznamů a po posledním měsíci je ze seznamů vytvořen výstup, který je buďto finální nebo je opět uložen do dalšího seznamu. Kromě ročního cyklu je zde i cyklus pro jednotlivá města, který funguje na podobném principu.

4.2 Zájmové území

Studie zkoumá efekt povrchových tepelných ostrovů měst v Evropě, proto bylo vybráno 10 reprezentativních měst, které mají různé vlastnosti a jsou rozprostřena na evropském kontinentu. Správný výběr zaručuje, že z výsledků lze vyvodit závěry pro celou Evropu.

Vybraná města: Amsterdam, Berlín, Budapešť, Londýn, Madrid, Paříž, Praha, Řím, Stockholm, Varšava



Obr. č. 4: Vybraná města ke studii (Zdroj: vlastní zpracování)

4.3 Identifikace městských a venkovských oblastí

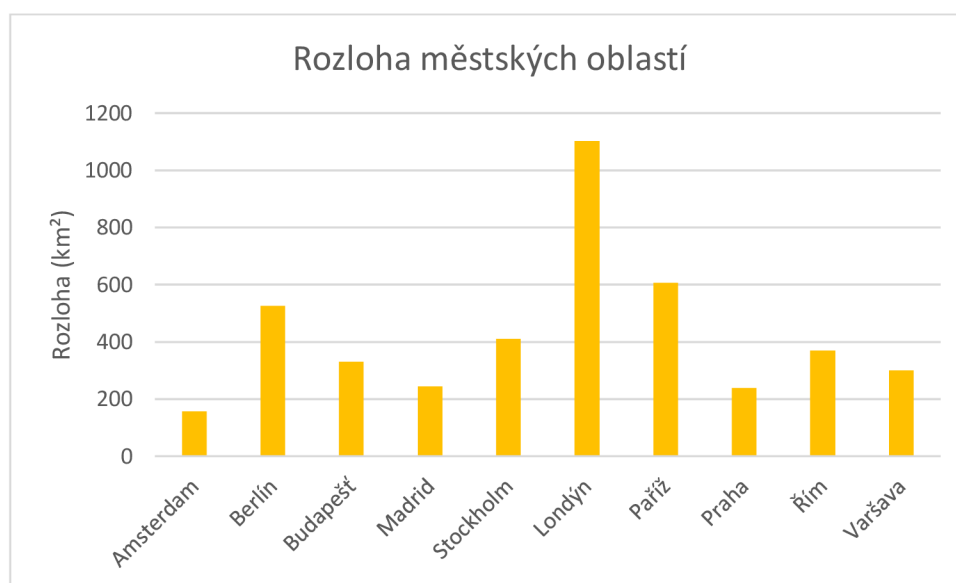
Prvním a jedním z nejdůležitějších kroků byla identifikace městských a venkovských oblastí. Zde byl využit software ArcGIS Pro 2.8.6 pro definování hranic a samotné povrchy byly identifikovány ve skriptech.

Nejdříve byly z portálu Copernicus Land Monitoring Service získány data Urban Atlas za rok 2018 pro každé město. Urban Atlas poskytuje data o krajinném pokryvu s vysokým rozlišením pro města, které mají více než 50000 obyvatel. Součástí balíčku dat je i datová sada s polygonem ohraničující městské jádro. Tyto polygony městského jádra byly využity jako hranice městských oblastí. V softwaru ArcGIS pak byly pro polygony městských oblastí vytvořeny buffery s průměrem 15 kilometrů. Tím vznikly polygony, které ohraničují venkovské oblasti. Ty byly následně nahrány na platformu Google Cloud, aby mohly být importovány do skriptů.

K určení městských a venkovských povrchů byla využita platforma GEE. Do skriptu byla importována data Copernicus CORINE Land Cover (CLC), která jsou dostupná v GEE katalogu. Konkrétně byla do skriptu importována rastrová datová sada CLC2018, která obsahuje informace o krajinném povrchu z roku 2018. Pro městské oblasti byly pomocí maskování vybrány pouze ty třídy, které označují umělé povrchy (třídy 111 až 124). Třídy (211 až 335), které označují přírodní povrchy (kromě

mokřadů a vodních ploch které by mohly ovlivnit přesnost SUHI) byly vybrány pro venkovské oblasti. Maskované třídy pak byly nakonec konvertovány z rastrového formátu na vektorový, aby mohly být později využity k ořezu snímků.

Pro odstranění vlivu nadmořských výšek, které by mohly potenciálně ovlivnit teploty byl využit digitální model terénu SRTM s rozlišením 30 metrů. Tento produkt byl do skriptu přidán z GEE katalogu. Nejdříve byly zjištěny maximální a minimální výšky městských oblastí. Poté byly z venkovských oblastí odstraněny oblasti, u kterých se rozdíl nadmořské výšky lišil s maximální nebo minimální výškou městské oblasti o více jak 30 metrů. Maskované oblasti pak byly konvertovány na vektorová data.



Obr. č. 5: Rozloha městských oblastí (km²) (Zdroj: vlastní zpracování)

4.3 Výpočet povrchové teploty a intenzity povrchových tepelných ostrovů měst

Výpočet povrchových teplot a intenzit povrchových tepelných ostrovů je další velmi významný krok. Tento krok byl podobný ve všech 3 skriptech, až na finální úpravy, které byly podle daného účelu řešeny v každém skriptu jinak.

Pro účely studie byly povrchové teploty získány z produktu MOD11A1. Tento produkt poskytuje každodenní měření noční a denní povrchové teploty s rozlišením 1 kilometru. Ty jsou odvozeny z produktu MOD11_L2 a obsahují pouze měření, které splňují podmínky jasné oblohy (Wan 2013).

Data byla nejdříve importována do skriptu z katalogu GEE. Poté byla Kolekce snímků vyříděna na denní a noční snímky z let 2015 až 2020. Dále byla kolekce

upravena na měsíční průměry povrchových teplot a přepočítána z kelvinů na stupně Celsia. Tyto snímky byly poté oříznuty městskými a venkovskými oblastmi, ze kterých byly následně vypočteny jejich průměrné číselné teploty.

Ve studii byla intenzita SUHI vypočtena jako rozdíl povrchových teplot v městských oblastech a povrchových teplot ve venkovských oblastech. Každý skript zkoumá intenzitu SUHI pro různé časové období a pracuje s různými typy objektů (snímek, číslo). Proto byly v jednotlivých skriptech využity různé postupy k získání SUHII:

Skript (a): V případě skriptu, který se zaměřuje na průzkum průměrných měsíčních pozorování SUHII v jednotlivých městech byly pro výpočet SUHII použity průměrné měsíční číselné hodnoty povrchové teploty v městských a venkovských oblastech.

Skript (b): Skript, který se zaměřuje na porovnání průměrných intenzit SUHI ve všech městech využívá k výpočtu SUHII číselný průměr všech měsíčních pozorování povrchových teplot pro dané město v městských a venkovských oblastech.

Skript (c): Ve skriptu, který se věnuje přípravě dat pro regresní analýzu je SUHII vypočítáno jako rozdíl průměrných měsíčních teplot v městské oblasti a průměrných číselných hodnot měsíčních teplot ve venkovské oblasti.

4.4 Výpočet řídicích faktorů

Pro studii bylo zvoleno 7 řídicích faktorů na základě literární rešerše. Pro tyto řídicí faktory byla vybrána vhodná data, která jsou až na nepropustné povrchy volně dostupná v katalogu GEE. Následně byla data zpracovaná ve skriptech GEE **(b)** a **(c)**.

Pro zkoumání hustoty a zdraví vegetace (NDVI) v městech a její interakci se SUHI byl zvolen produkt MOD13A2.006 Terra Vegetation Indices 16 – Day Global 1 km. To je produkt generovaný pomocí osmidenních údajů o odrazivosti. Produkt poskytuje dva vegetační indexy: NDVI a EVI s rozlišením 1 kilometru. Algoritmus tohoto produktu vybere nejlepší pixel z periody 16 dní, kde je nízké zastoupení oblačnosti a nejvyšší hodnota NDVI/EVI (Didan et al., 2015).

Další zkoumaný faktor byl obsah vody (NDWI) ve městech a její vliv na SUHI. Produkt, který byl v práci využit poskytuje denní hodnoty NDWI s rozlišením 500

metrů, které jsou generovány z odrazivosti povrchu produktu MOD09GA (Vermote et al., 2015).

Také bylo zkoumáno albedo povrchů pomocí produktu MCD43A3, který poskytuje data s rozlišením 500 metrů. Ten je generován pomocí 16denních dat z družic Terra i Aqua a poskytuje albedo při difuzním ozáření (WSA) a albedo při přímém ozáření (BSA) viditelného, blízkého infračerveného a krátkého infračerveného pásma MODIS (Schaaf, 2022). Protože albedo bílé oblohy vykazuje lineární korelaci s albedem černé oblohy a mají podobný vlivu na SUHI (Zhou et al., 2014) bylo v této studii použito pouze albedo bílé oblohy.

Dalším zkoumaným faktorem bylo záření světla. K tomu byl využit produkt VIIRS Stray Light Corrected Nighttime Day/Night Band Composites Version 1 s rozlišením 500 metrů. Ten kombinuje snímky s průměrným měsíčním zářením světla, které využívají noční pozorování z Visible Infrared Imaging Radiometer Suite (VIIRS), respektive jeho denní/noční pásma (DNB) (Mills et al., 2013).

Data byla postupně naimportována do skriptu a vytříděna na denní a noční kolekce snímků z let 2015 až 2020. Tyto kolekce pak byly upraveny na měsíční průměry a oříznuty polygony městských oblastí.

Studie se také věnovala průzkumu počtu obyvatel v městských oblastech a vlivu hustoty obyvatel na SUHI. K tomu byl využit produkt WorldPop Global Project Population Data: Estimated Residential Population per 100x100m Grid Square. Projekt WorldPop poskytuje podrobná globální otevřená data s vysokým rozlišením o distribuci lidské populace. Tato datová sada odhaduje počet lidí žijících v jednotlivých buňkách (WorldPop, 2022). Tyto data jsou poskytována ročně, proto zde byl použit jiný postup. Po importování dat do skriptu byla data upravena na snímek s průměrnými počty obyvatel za roky 2015 až 2020, který byl následně oříznut polygony městských oblastí.

Jediný faktor, pro který katalog neposkytoval žádné produkty byly nepropustné povrchy. Proto byl tedy k průzkumu efektu nepropustných povrchů zvolen volně dostupný produkt Imperviousness Density 2018. Tento produkt je spravován Evropskou agenturou pro životní prostředí v rámci programu EU Copernicus. Vyjadřuje procentuální stupeň nepropustnosti povrchu za období 2017 až 2019 s rozlišením 10 metrů. Nepropustnost je získána pomocí poloautomatické klasifikace kalibrovaného NDVI. Produkt je vytvořen pro oblasti, kde se nachází zastavěné plochy (European Environment Agency, 2022). Data byla stažena z portálu Copernicus Land Monitoring Service pro každé město zvlášť, následně byla otevřena

v softwaru ArcGIS Pro. Nejdříve bylo potřeba identifikovat rastry, které pokrývají zvolená města. Vybrané rastry byly spojeny do mozaiky a vyexportovány. Nakonec byly data nahrána na platformu Google Cloud pro další práci v GEE.

Dále byl zkoumán vliv nadmořských výšek, k tomu byla použita data o nadmořských výškách z digitálního modelu terénu NASA Shuttle Radar Topography Mission (SRTM) Digital Elevation 30 m. Tento produkt 3. verze je poskytován NASA s rozlišením 1 obloukové sekundy (asi 30 metrů) (Farr et al., 2007).

Tato data potřebovala pouze naimportovat do skriptu a oříznout polygony městských oblastí.

V dalším kroku byly ze všech snímků vybrány pouze ty pásma, které obsahovaly požadované hodnoty, tzn.: NDVI, elevation, population, NDWI, avg rad, Albedo WSA shortwave a b1 (nepropustné povrchy). Takto zpracované snímky byly dále využity ve dvou skriptech:

Skript (b): Z průměrných měsíčních snímků byly vypočteny průměrné měsíční číselné hodnoty pixelů (kromě snímků s počty obyvatelstva, u kterých byly hodnoty pixelů sečteny) a z těchto hodnot pak byla vypočtena průměrná číselná hodnota pro jednotlivá města za celé období.

Skript (c): Z kolekce průměrných měsíčních hodnot snímků byly vypočteny průměrné snímky jednotlivých měst s hodnotami za celé období.

4.5 Vytvoření grafických a datových výstupů

Finálním krokem v rámci GEE skriptů bylo vytvoření vhodných grafických výstupů, které by vhodně reprezentovaly vývoj a trend SUHI ve městech a průměrné hodnoty jejich řídicích faktorů. Také byl vytvořen bodový výstup se vzorky pro regresní analýzu.

Vzhledem k tomu, že jednotlivé skripty slouží k různým účelům a liší se postupem, bylo potřeba vypracovat grafické a datové výstupy zvlášť.

Skript (a): V tomto skriptu byly noční a denní měsíční průměry SUHII postupně přidány do sbírek (arrays), po dokončení ročního cyklu pro jednotlivá města byly ze sbírek nočních SUHII a denních SUHII vytvořeny grafy pro každé město, které znázorňují noční nebo denní měsíční vývoj SUHII ve městech.

Skript (b): Tento skript postupně přidává číselné měsíční průměrné hodnoty řídicích faktorů do sbírek, které jsou na konci ročního skriptu zprůměrovány pro jednotlivá města a přidány do nové sbírky. Po dokončení cyklu ve všech městech je z každé sbírky vytvořen graf, který znázorňuje průměrné hodnoty řídicího faktoru v každém městě.

Skript (c): V posledním skriptu byly průměrné měsíční snímky řídicích faktorů, nočních a denních SUHII přidány do sbírek. Na konci ročního cyklu byly také přidány snímky řídicích faktorů, pro které nebyly vytvořeny měsíční průměry a také polygony městské oblasti. Po dokončení cyklu ve všech městech, byly z jednotlivých sbírek, které obsahovaly měsíční průměry, vytvořeny průměrné hodnoty snímků. Ze sbírek, které neobsahují měsíční snímky byly vytvořeny mozaiky, a ve sbírce s městskými oblastmi byly oblasti spojené dohromady. Dále byly snímky s průměrnými hodnotami v městských oblastech přidány jako pásma do jednoho společného snímku. V městských oblastech byl pak vytvořen určitý počet náhodně rozmístěných vzorků (bodů), které byly od sebe vzdálené 1,5 kilometru a obsahovaly pouze hodnoty, které nabývaly všechny pásma ve snímku. Z těchto vzorků byla následně vytvořena tabulka, který byla nahrána na Google Drive.

4.6 Regresní analýza

V poslední části byla provedena regresní analýza SUHII a jeho řídicích faktorů. Účelem analýzy bylo zjistit vhodnost zvolených řídicích faktorů pro vysvětlení SUHII a vliv jednotlivých faktorů. Postup byl proveden v programu IBM SPSS 28.0.1 a tabulky a grafy byly vytvořeny v programu Excel.

V první fázi byly z Google Drive staženy tabulky s naměřenými hodnotami, které byly vytvořeny skriptem v GEE. Tyto tabulky byly nahrány do programu SPSS. Následně byly z každé sady vzorků vytvořeny modely lineární mnohonásobné regrese pro noční SUHII a denní SUHII.

Jako vstupní závislá proměnná byl zvolen faktor SUHI (denní a noční), ostatní faktory byly zvoleny jako nezávislé proměnné. Pro nalezení nejlepšího modelu byla zvolena metoda Stepwise, která vybere do modelu pouze signifikantní proměnné s pravděpodobností F pro přidání proměnné 0,05 a pro odebrání proměnné 0.1. Pro regresní analýzu byly zohledněny tyto podmínky:

- 1) Závisle proměnná je měřena na intervalové úrovni.
- 2) Nezávislé proměnné jsou měřené na intervalové úrovni.

- 3) Autokolinearita byla řešena vzdáleností 1,5 km mezi vzorky.
- 4) Multikolinearita – proměnné mezi sebou nevykazují větší korelační koeficient než 0,9, ukazatel tolerance je vždy větší než 0,2, ukazatel VIF je vždy menší než 5.

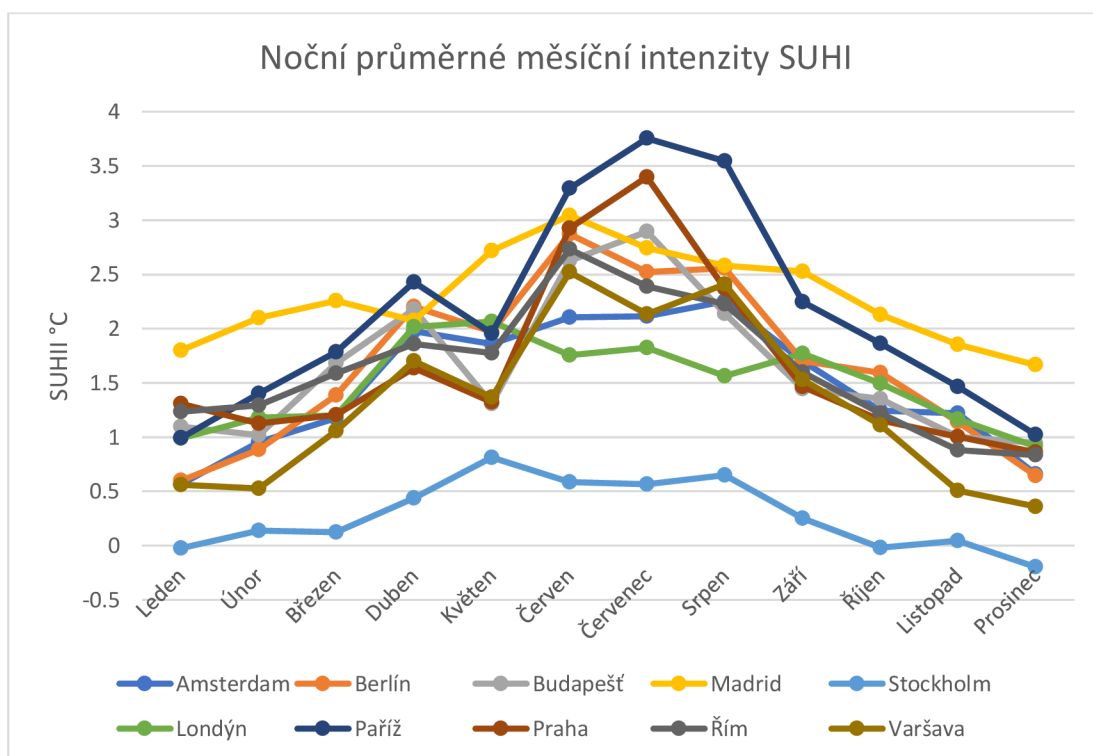
Po spuštění procesu byl vždy vybrán “nejlepší” model, ze kterého byly vytvořeny výstupní tabulky a grafy.

5. Výsledky

5.1 Intenzity povrchových tepelných ostrovů

5.1.1 Noční intenzita povrchového tepelného ostrova

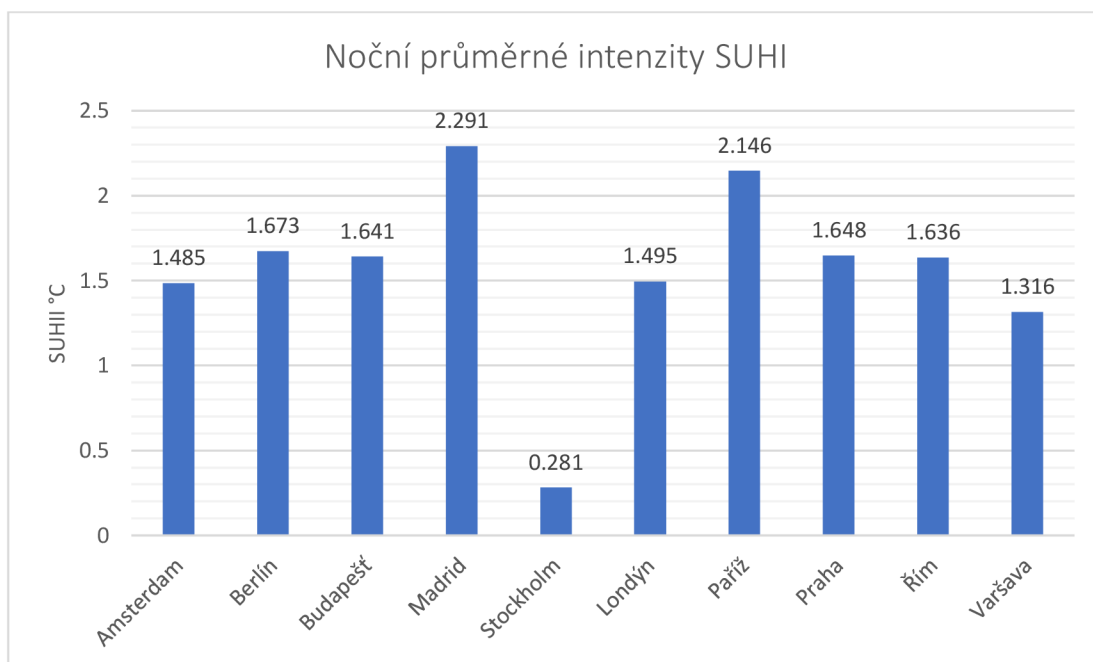
Jak lze vypočítat z grafu (obr. č. 6), intenzity jsou během roku velmi proměnlivé. Nejvyšších intenzit dosahují města v letních měsících. Paříž dosahuje nejvyšší intenzity ze všech měst (3,754 °C) v červenci. Praha je město s druhou nejvyšší intenzitou (3,398 °C) také v červenci. Nejnižší SUHI se vyskytují v zimních měsících s tím, že Stockholm se vymyká ostatním městům a dosahuje výrazně nižších intenzit, jeho nejnižší intenzita se objevuje v prosinci (-0,197 °C).



Obr. č. 6: Noční průměrné měsíční intenzity SUHI (Zdroj: vlastní zpracování)

Podle grafu (obr. č. 7) dosahuje nejvyšší noční průměrné intenzity SUHI Madrid (2,291 °C). Velmi podobně je na tom Paříž, která s průměrnou intenzitou

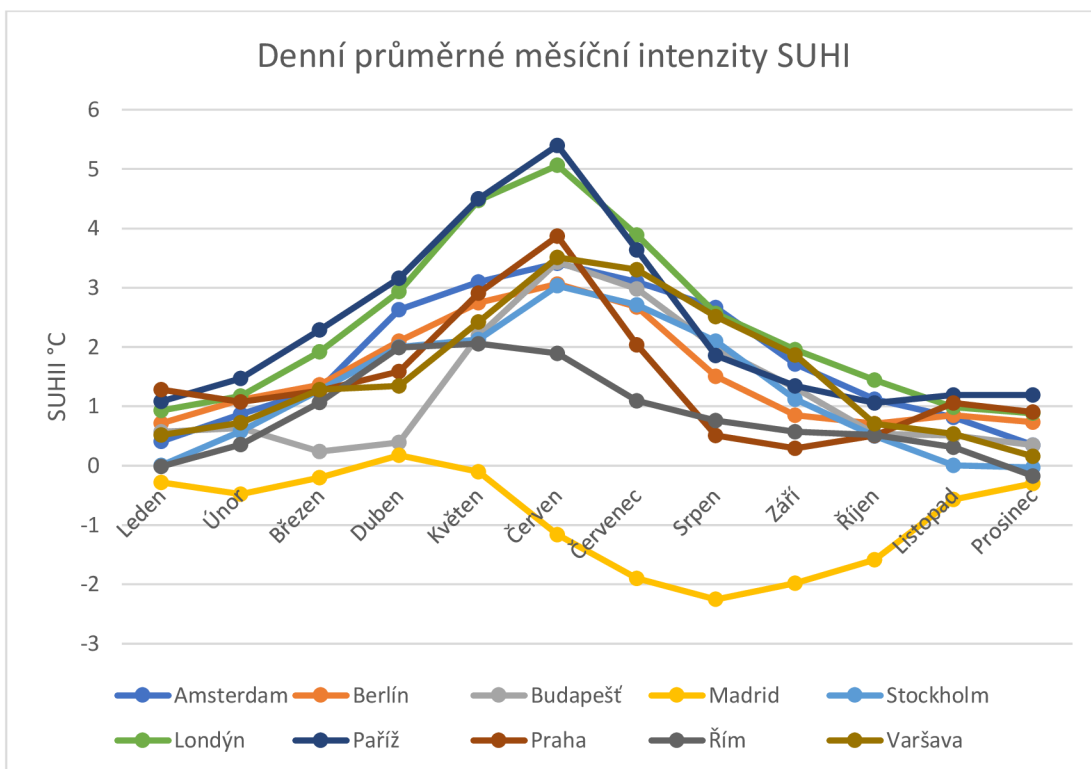
(2,146 °C) dosahuje druhé nejvyšší intenzity. Ostatní města dosahují podobných hodnot (kolem 1,5 °C) s výjimkou Stockholmu, který vykazuje velmi malou průměrnou intenzitu SUHI (0,281 °C).



Obr. č. 7: Průměrné noční intenzity SUHI (Zdroj: vlastní zpracování)

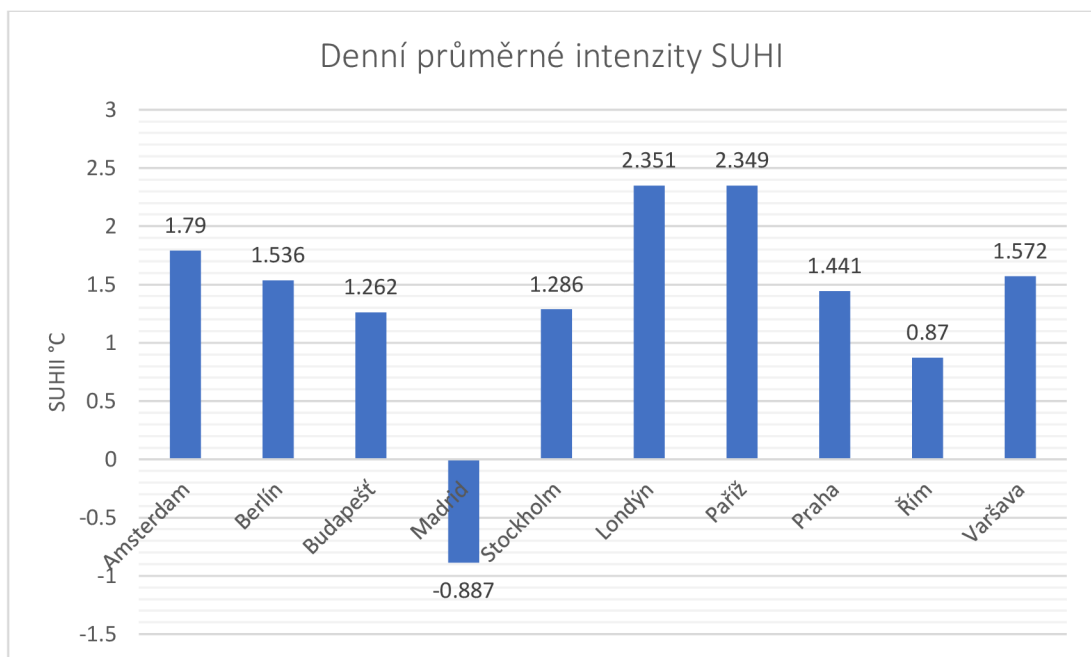
5.1.2 Denní intenzita povrchového tepelného ostrova

Z grafu (obr. č. 8) lze vyčíst, že nejvyšší intenzity se vyskytují v červnu. Paříž dosahuje nejvyšší intenzity (5,4 °C) a velmi podobné intenzity dosahuje také Londýn (5,06 °C). Nejnižší intenzity se objevují v zimních měsících, ve kterých města dosahují podobných intenzit (kolem 0,5 °C). Madrid se významně odlišuje od ostatních měst, má odlišný průběh a většinu měsíců se pohybuje v negativních intenzitách. Nejnižší intenzita Madridu se objevuje v srpnu (-2,54 °C), což je i nejnižší intenzita ze všech měst.



Obr. č. 8: Denní průměrné měsíční intenzity SUHI (Zdroj: vlastní zpracování)

Graf (obr. č. 9) je výsledky velmi podobný předchozímu grafu (obr. č. 8). Nejvyšší průměrné intenzity zde dosahuje Londýn (2,351 °C) a Paříž má jen nepatrně menší intenzitu (2,349 °C). Výrazně nejnižší intenzita se vyskytuje v Madridu (-0,887 °C),



Obr. č. 9: Denní průměrné intenzity SUHI (Zdroj: vlastní zpracování)

5.2. Analýza řídicích faktorů SUHI

5.2.1 Průměrné hodnoty řídicích faktorů v městských oblastech

Z tabulky (tab. č. 1) lze zjistit, že nejvyšší průměrné NDVI dosahuje Londýn, nejvyšší průměrné NDWI Stockholm a nepropustnost Paříž. Dále je z tabulky zřejmé, že Madrid dosahuje nejvyšší průměrnou nadmořskou výšku, hustotu obyvatel, záření světél a albedo.

Nejnižší průměrné NDVI má Madrid, Amsterdam má nejnižší průměrnou nadmořskou výšku, Stockholm nejnižší hustotu obyvatel a nepropustnost povrchu, záření světél je nejnižší v Berlíně a albedo je nejnižší v Amsterdamu.

Metropole	NDVI	Nadmořská výška (m)	Hustota obyvatel (100 m ²)	NDWI	Záření světél (Nano watt/cm ² /sr)	Albedo (WSA)	Nepropustnost (%)
Amsterdam	0.429	1.469	28.25	0.01	29.239	0.114	59.359
Berlín	0.461	44.456	26.73	0.01	18.511	0.125	49.994
Budapešť	0.412	137.184	28.45	0.00	27.324	0.134	40.169
Madrid	0.281	653.584	77.84	-0.03	55.341	0.164	57.763
Stockholm	0.43	35.612	14.55	0.02	29.997	0.122	31.601
Londýn	0.479	40.331	41.50	0.01	32.396	0.122	52.714
Paříž	0.377	64.094	62.73	0.00	52.942	0.132	68.231
Praha	0.449	275.279	21.53	0.01	28.387	0.143	56.208
Řím	0.447	53.821	42.75	0.00	41.82	0.146	49.138
Varšava	0.414	95.146	25.27	0.01	39.251	0.13	38.013

Tab. č. 1: Průměrné hodnoty řídicích faktorů (Zdroj: vlastní zpracování)

Zelená barva = nejnižší hodnoty

Tučně = nejvyšší hodnoty

5.2.2 Výsledky vícenásobné lineární regrese

Vysvětlované modely obsahují pouze ty proměnné, které nebyly vyřazeny metodou Stepwise (viz. Kapitoly: 3.7.1 Metoda Stepwise a 4.6 Regresní analýza).

Vhodnost modelu

U všech vypočtených koeficientů determinace byla ověřena platnost nulové hypotézy ($R^2 = 0$), tedy že nelze vysvětlovat SUHI pomocí zvolených řídicích faktorů metodou Stepwise. Na koeficientech byl proveden F test a následně byla ověřena jeho signifikance. Lze tedy říct, že vypočtené koeficienty determinace jsou statisticky

signifikantní na hladině 95 %. To značí, že se SUHII dá alespoň částečně vysvětlit pomocí zvolených řídicích faktorů.

Z tabulky č.2 lze vyčíst, že regresní modely vysvětlující variabilitu nočních intenzit SUHI jsou vhodnější než modely vysvětlující variabilitu denních intenzit SUHI. Model, který vysvětluje variabilitu nočních intenzit SUHI v Římě se jeví jako nejvhodnější ($R^2 = 0,872$ – nezávisle proměnné vysvětlují 87,2 % variability v závisle proměnné). Model vysvětlující variabilitu denních intenzit SUHI v Amsterdamu se jeví jako nejhorší ($R^2 = 0,428$ - nezávisle proměnné vysvětlují 42,8 % variability v závisle proměnné).

Metropole	Závisle proměnná	Přizpůsobené R^2
Amsterdam	SUHII denní	0.428
	SUHII noční	0.605
Berlín	SUHII denní	0.745
	SUHII noční	0.65
Budapešť	SUHII denní	0.608
	SUHII noční	0.623
Londýn	SUHII denní	0.44
	SUHII noční	0.451
Madrid	SUHII denní	0.708
	SUHII noční	0.81
Paříž	SUHII denní	0.621
	SUHII noční	0.664
Praha	SUHII denní	0.617
	SUHII noční	0.82
Řím	SUHII denní	0.467
	SUHII noční	0.872
Stockholm	SUHII denní	0.677
	SUHII noční	0.48
Varšava	SUHII denní	0.557
	SUHII noční	0.826

Tab. č. 2: Hodnoty koeficientů determinace (Zdroj: vlastní zpracování)

Zelená barva = nejnižší hodnoty

Tučně = nejvyšší hodnoty

V průměru se podařilo vysvětlit 68% variability noční SUHII a 58% denní SUHII.

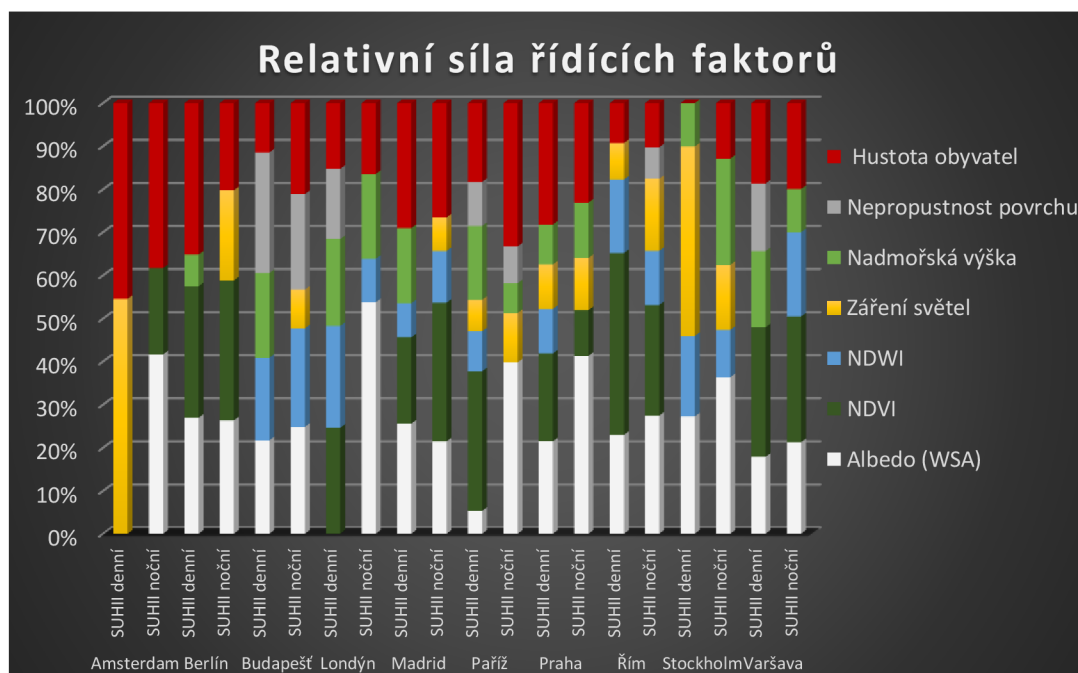
Relativní vliv a vztah řídicích faktorů na SUHII

Z tab. č. 3 lze z hodnot regresních koeficientů určit vztah mezi jednotlivými faktory a noční nebo denní SUHII. Kladná hodnota indikuje pozitivní vztah a záporná hodnota indikuje negativní vztah. Lze tedy říct, že albedo má negativní vztah se SUHII v noci a pozitivní ve dne. NDVI má negativní vztah se SUHII ve dne i v noci, stejně

jako nadmořská výška. NDWI má pozitivní vztah se SUHII v noci a negativní ve dne. Záření světla má pozitivní vztah s nočním i denním SUHII, stejně tak i nepropustnost povrchu a hustota obyvatel.

Příloha č. 4 a obr. č. 10 nám pomáhají určit vliv jednotlivých faktorů na denní a noční SUHII ve zkoumaném městě. Z absolutních hodnot regresních koeficientů lze vyčíst sílu nezávisle proměnné v modelu a tím pádem i vliv neboli schopnost faktorů vysvětlovat denní a noční SUHII.

NDWI a nadmořská výška mají velmi malý vliv na denní a noční SUHII a nedominují ani v jednom modelu. Nepropustnost povrchu má výraznější vliv na denní SUHII pouze v Budapešti, v ostatních modelech je jeho vliv velmi malý. Záření světla dominuje u denních intenzit SUHI v Amsterdamu a Stockholmu, ale jinak je v ostatních modelech vliv faktoru poměrně malý. Vliv hustoty obyvatel je oproti předchozím faktorům relativně větší, a to především v denních modelech intenzit SUHI v Berlíně, Madridu a Prahy, kde je dominantním faktorem. Jako nejvlivnější faktor v modelech nočních SUHII se jeví albedo, které dominuje v 7 modelech, konkrétně v Amsterdamu, Budapešti, Londýně, Paříži, Prahy, Římě a Stockholmu. Jako další významně vlivný faktor se jeví NDVI, které je nejvlivnější především ve 4 modelech denních SUHII (Londýn, Paříž, Řím, Varšava), ale také ve 3 modelech nočních SUHII (Berlín, Madrid, Varšava).



Obr. č. 10: Relativní síla řídicích faktorů (Zdroj: vlastní zpracování)

Model		standardizované regresní koeficienty						
		Albedo (WSA)	NDVI	NDWI	Záření světél	Nadmořská výška	Nepropustnost povrchu	Hustota obyvatel
Amsterdam	SUHII denní	*	*	*	0.406	*	*	0.338
	SUHII noční	-0.574	0.277	*	*	*	*	0.528
Berlín	SUHII denní	0.33	-0.373	*	*	-0.09	*	0.43
	SUHII noční	-0.335	-0.413	*	0.266	*	*	0.257
Budapešť	SUHII denní	0.302	*	-0.266	*	-0.274	0.389	-0.159
	SUHII noční	-0.304	*	0.28	0.11	*	0.271	0.258
Londýn	SUHII denní	*	0.409	-0.393	*	-0.334	0.271	0.253
	SUHII noční	-0.491	*	0.092	*	-0.18	*	0.15
Madrid	SUHII denní	0.315	-0.247	-0.097	*	-0.214	*	-0.357
	SUHII noční	-0.351	-0.526	0.196	-0.127	*	*	0.434
Paříž	SUHII denní	0.083	-0.508	-0.146	0.114	-0.268	0.161	-0.286
	SUHII noční	-0.415	*	*	0.12	-0.072	0.089	0.347
Praha	SUHII denní	0.34	-0.322	-0.164	0.163	-0.146	*	0.446
	SUHII noční	-0.496	-0.128	*	0.145	-0.155	*	0.277
Řím	SUHII denní	0.517	-0.948	0.384	-0.19	*	*	-0.208
	SUHII noční	-0.37	-0.346	0.169	0.226	*	0.098	0.138
Stockholm	SUHII denní	0.325	*	-0.223	0.524	-0.12	*	*
	SUHII noční	-0.483	*	-0.147	0.2	-0.328	*	0.171
Varšava	SUHII denní	0.205	-0.344	*	*	0.202	0.18	0.213
	SUHII noční	-0.317	-0.434	0.29	*	0.15	*	0.296

Tab. č. 3: Standardizované regresní koeficienty (Zdroj: vlastní zpracování)

* = proměnná byla vyřazena z modelu

6. Diskuse

6.1 Zhodnocení metodiky

Řídící faktory, které se objevují v bakalářské práci byly zvoleny na základě předchozích studií a dostupnosti dat. Zde je nutno podotknout, že řídicích faktorů, které působí na intenzitu SUHI je mnohem více, například Wang et al. (2021) ve své práci zkoumali mimo jiné i vliv vzdálenosti od vodních ploch a cest, krajinného pokryvu nebo hrubého domácího produktu. V určeném rozsahu práce ale nebylo možné zahrnout vliv všech dostupných řídicích faktorů.

Co se týče identifikace městských a venkovských oblastí, existuje zde několik technik a názorů, jak přesně by měly být tyto oblasti identifikovány. V této práci byly oblasti identifikovány pomocí krajinného pokryvu a rozdílu v nadmořských výškách, ale například Martin-Vide et al. (2015) tvrdí, že venkovské oblasti by měly mít vzdálenost 800 metrů od všech vodních ploch, to by ale v této práci nebylo vhodné vzhledem k tomu, že data Corine Land Cover nejsou příliš podrobná. Zhou et al. (2014) použili k identifikaci městských oblastí intenzitu zástavby vygenerovanou z krajinného pokryvu a k identifikaci venkovských oblastí použili buffer kolem městské oblasti se stejnou rozlohou jako rozloha městské oblasti. Na podobném postupu pro identifikaci venkovských oblastí se shoduje více studií, Dewan et al. (2021) také konstatuje, že rozloha venkovských oblastí by měla být podobná jako rozloha městských oblastí. V této práci byl použit pro všechny města 15kilometrový buffer bez ohledu na rozlohu městských oblastí, z obrázku č.5 lze vyzorovat, že rozlohy městských oblastí jsou až na pár výjimek poměrně výrazně rozdílné. Z tohoto důvodu by v budoucnu bylo lepší přizpůsobit rozlohy venkovských oblastí rozlohám městských oblastí.

Jako velmi důležité se jeví správné provedení regresní analýzy. Ve studii byl zvolen model vícenásobné lineární regrese, do tohoto modelu by měly vstupovat data, které splňují určité podmínky. Splněné podmínky jsou vypsány v metodické části, v podkapitole 4.5 Regresní analýza. Jedna z podmínek, která byla splněna jen částečně je, že by proměnné měly být v lineárním vztahu. Ve studii byla linearita předpokládána především na základě předchozích studií, které zkoumaly podobné faktory pomocí lineární regrese (Zhou et al., 2014). Také byla linearita částečně prověřena z korelačních diagramů, ale vzhledem k nezkušenosti, nelze s jistotou říci, že byla linearita dodržena. Další podmínka, která nebyla ověřena je výskyt odlehlých hodnot. Tato podmínka nebyla nijak ověřena a měla být. Bylo pouze předpokládáno, že vzhledem k faktu, že data byla průměrným pozorování za 5 let, nemusely by se

zde odlehle hodnoty vyskytovat. Neověřená zůstala také podmínka normality a podmínka homogenity rozptylu. Důležité je také podotknout, že není zaručeno, že do modelu vždy vstupuje adekvátní počet vzorků. Jejich počet se odvíjí od rozlohy městských oblastí, neboť od sebe musí být vzorky 1,5 kilometru, aby nedocházelo k autokorelaci.

6.2 Zhodnocení a porovnání výsledků

6.2.1 Intenzita SUHI ve městech

Při zkoumání intenzit povrchových tepelných ostrovů ve městech se potvrdilo, že efekt denního i nočního povrchového ostrova se vyskytuje ve všech zkoumaných městech. Intenzity SUHI byly obecně vyšší ve dne než v noci. To souhlasí s výsledky Dewan et al. (2021) a Peng et al. (2011), kteří došli ke stejnému závěru, ale naopak oponuje výsledkům Zhou et al. (2014), kteří došli k opačným výsledkům. Dále bylo prokázáno, že se intenzity velmi liší v závislosti na ročním období, což potvrzuje i Zhou et al. (2014). V letních měsících se vyskytovaly nejvyšší denní i noční intenzity. Nejnižší intenzity byly v obou případech nalezeny v zimních měsících s výjimkou Madridu, který vykazoval negativní intenzity v průběhu celého roku, takovýto trend SUHI potvrzuje i Stathpoulou & Cartalis (2007). Zajímavé také je, že noční SUHI byly nejvyšší v červenci a denní SUHI v červnu.

Výrazně odlišné chování oproti ostatním městům vykazovaly Madrid a Stockholm. Madrid se projevoval negativními intenzitami SUHI ve dne, a naopak velmi vysokými intenzitami SUHI v noci. Stockholm měl velmi nízké intenzity v noci, ale jeho intenzity SUHI ve dne byly podobné jako u ostatních měst. Takové chování je spíše vzácné a objevuje se ve městech se specifickými podmínkami, to dokazuje Peng et al. (2011), kteří provedli studii na 419 městech a 8 % z nich dosahovaly negativní intenzity SUHI. Negativní intenzity SUHI v Madridu zjistili i Sobrino et al. (2012) a potvrzují tak neobvyklé výsledky nalezené v této práci. Můj odhad je, že se ve venkovských oblastech Madridu vyskytuje velký poměr holé půdy a malý poměr zdravé vegetace s dostatkem vody. To způsobuje vysoké teploty povrchu při přímém vystavení slunečnímu záření a v Madridu tak vzniká městský chladný ostrov. V noci se pak půda rychle ochladí, neboť nedokáže zadržet teplo na delší dobu a Madrid je vystavován efektu městského tepelného ostrova.

6.2.2 Vícenásobná lineární regrese

Vzhledem k výsledkům lineární regrese se dá říct, že vypočítané regresní modely mají poměrně dobrou míru kvality, tedy vysvětlují poměrně velkou část

variace závisle proměnných a stejně tak jako ve studii Zhou et al. (2014) vysvětlují faktory lépe variance nočních intenzit SUHI (průměrné noční SUHI = 68 %, průměrné denní SUHI = 58 %). To indikuje, že pro denní SUHI zde existuje komplexnější mechanismus a v budoucnu je potřeba zkoumat vliv dalších faktorů, které by denní intenzity SUHI vysvětlovaly lépe.

Co se týče vlivu jednotlivých faktorů, jejich vliv byl různý v závislosti na zkoumaném městě a času pozorování SUHI. Při porovnávání jejich vlivu lze říct, že na noční intenzity SUHI má největší vliv albedo (průměrná absolutní hodnota standardizovaného $B = 0,414$), které dominovalo v 7 městech a na denní SUHI má největší vliv NDVI (průměrná absolutní hodnota standardizovaného $B = 0,315$), které dominovalo ve 4 městech. Za zmínku stojí také faktor hustoty obyvatel, který měl největší vliv na denní SUHI ve 3 městech (průměrná absolutní hodnota standardizovaného $B = 0,286$). Ve studii Zhou et al. (2014) měly na denní SUHI největší vliv antropogenní faktory a podnebí. Na noční SUHI mělo v jejich studii největší vliv albedo, stejně jako v mé práci. Wang et al. (2021) zkoumali vliv faktorů na celkové průměrné SUHI a v jejich výsledcích měly největší vliv noční záření světla a nepropustné povrchy. Na druhou stranu v práci Peng et al. (2011) vyhodnotili jako nejdůležitější faktory albedo a záření světla pro noční SUHI a vegetaci pro denní SUHI. Z toho lze usoudit, že výsledky jsou poměrně proměnlivé a není zde jasně daný faktor, který by ovlivňoval SUHI. Obecně se ale dá říct, že pro obranu proti nočnímu SUHI je potřeba zvýšit ve městě hodnoty albeda a proti dennímu SUHI vysázet více zeleně. Každé město by si ale mělo opatřit své vlastní plány a doporučení a řídit se především jimi.

7. Závěr a přínos práce

V této práci byly zkoumány noční a denní intenzity povrchových tepelných ostrovů a jejich řídicí faktory v 10 evropských metropolích (Amsterdam, Berlín, Budapešť, Londýn, Madrid, Paříž, Praha, Řím, Stockholm, Varšava) v období 2015-2020. Výběr měst byl řízen především tím, aby města reprezentovala většinu měst v Evropě a řídicí faktory byly zvoleny na základě literární rešerše a dostupnosti dat.

Bakalářská práce byla převážně zpracována na platformě Google Earth Engine. V rámci této platformy bylo vypracováno několik skriptů, které zpracují vstupní data a vytvoří potřebné grafické a datové výstupy, které jsou dále zpracovány nebo přímo interpretovány. Výhoda takového postupu přináší možnost úpravy jakékoli části skriptu bez toho, aniž by se celý postup musel manuálně přepracovávat.

Při zkoumání intenzit povrchových tepelných ostrovů ve městech se potvrdily předpoklady z literární rešerše. Fenomén SUHI se objevil ve všech zkoumaných městech a jeho intenzity byly vyšší ve dne než. Nejnižší intenzity byly nalezeny v zimních měsících a nejvyšší v letních měsících. Velmi odlišně se intenzity SUHI projevovaly v Madridu, kde během dne dosahovaly záporných hodnot a během noci byly naopak velmi vysoké.

Důležitou součástí práce bylo zkoumání vlivu řídicích faktorů na noční a denní intenzitu SUHI. K tomu byl zvolen model vícenásobné lineární regrese a metoda stepwise. Vypočítané modely měly poměrně dobrou míru kvality s tím, že faktory vysvětlovaly lépe noční SUHI než denní SUHI. Na noční intenzity SUHI mělo největší vliv albedo. NDVI pak mělo největší vliv na denní intenzity SUHI.

Evropská města by se měla především zaměřit na větší pokrytí měst vegetací a na správný výběr materiálu při výstavbě. Obzvláště velký důraz na tyto dva faktory by měly dbát v Madridu, Londýně a Paříži, kde dochází k vysokým intenzitám SUHI. Výsledky v této práci byly ověřeny a porovnány s výsledky jiných autorů a dá se říct, že dosažené výsledky splňují očekávání.

8. Přehled literatury a použitých zdrojů

Aguilar-Lome J., Espinoza-Villar R., Espinoza J.-C., Rojas-Acuña J., Willems B. L., Leyva-Molina W.-M., 2019: Elevation-dependent warming of land surface temperatures in the Andes assessed using MODIS LST time series (2000–2017). *International Journal of Applied Earth Observation and Geoinformation* 77: 119–128.

Arellano B. & Roca J., 2021: REMOTE SENSING AND NIGHT TIME URBAN HEAT ISLAND. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 43: 15-22.

Arrau P. C., Peña A. M., 2021: The Urban Heat Island (UHI) Effect (online) [cit. 2022.03.05], dostupné z <<https://www.urbanheatlands.com/home>>.

Bahi H., Radoine H., Mastouri H., 2019: Urban Heat Island: State of the Art. 2019 7th International Renewable and Sustainable Energy Conference (IRSEC), Morocco, 7 s.

Basu A., Das S., 2021: Afforestation, revegetation, and regeneration: a case study on Purulia district, West Bengal (India). *Modern Cartography Series* 10: 497-524.

Buchin O., Hoelscher M.-T., Meier F., Nehls T., Ziegler F., 2016: Evaluation of the health-risk reduction potential of countermeasures to urban heat islands. *Energy and Buildings* 114: 27–37.

Casandra: What Is the Heat Island Effect? (online) [cit.2022.03.01.], dostupné z <<http://www.gardinergreenribbon.com/heat-island-effect>>.

- Dewana A., Kiseleva G., Botjea D., Mahmud I. G., Bhuiyan Md. H., Hassan Q. K., 2021: Surface urban heat island intensity in five major cities of Bangladesh: Patterns, drivers and trends. *Sustainable Cities and Society* 71: 102926.
- Didan K., Munoz B. A., Solano R., Huete A., 2015: MODIS Vegetation Index User's Guide (MOD13 Series). The University of Arizona, 32 s.
- Earth Resources Observation and Science (EROS) Center, ©2019: Urban Heat Islands (online) [cit.2022.03.02.], dostupné z <<https://www.usgs.gov/media/images/urban-heat-islands>>.
- European Environment Agency, ©2022: Copernicus Programme (online) [cit.2022.03.13.], dostupné z <<https://land.copernicus.eu/>>.
- Farr T. G., Rosen P. A., Caro E., Crippen R., Duren R., Hensley S., Hensley S., Kobrick M., Paller M., Rodriguez E., Roth L., Seal D., Shaffer S., Shimada J., Umland J., Werner M., Oskin M., Burbank D., Alsdorf, D., 2007: The Shuttle Radar Topography Mission. *Reviews of Geophysics* 45(2).
- Foster E., 2020: Four Approaches to Reducing the Urban Heat Island Effect (online) [cit.2020.03.02.], dostupné z <<https://urbanland.uli.org/sustainability/four-approaches-to-reducing-the-urban-heat-island-effect/>>.
- Gandhi S.M., Sarkar B.C., 2016: Reconnaissance and Prospecting. *Essentials of Mineral Exploration and Evaluation*: 53-79.
- Gao B., 1996: NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sensing of Environment* 58(3): 257–266.
- Hu Y., Dai Z., Guldmann J.-M., 2020: Modeling the impact of 2D/3D urban indicators on the urban heat island over different seasons: A boosted regression tree approach. *Journal of Environmental Management* 266: 110424.
- Khare V., Vajpai A., Gupta D., 2021: A big picture of urban heat island mitigation strategies and recommendation for India. *Urban Climate* 37: 100845.
- Li L., Yu T., Zhao L., Zhan Y., Zheng F., Zhang Y., Mumtaz F., Wang, C., 2019: Characteristics and trend analysis of the relationship between land surface temperature and nighttime light intensity levels over China. *Infrared Physics & Technology* 97: 381–390.
- Martin-Vide J., Sarricolea P., Moreno-García M. C., 2015: On the definition of urban heat island intensity: the “rural” reference. *Frontiers in Earth Science* 3: 24.
- Mathew A., Sarwes P., Khandelwal S., 2022: Investigating the contrast diurnal relationship of land surface temperatures with various surface parameters represent vegetation, soil, water, and urbanization over Ahmedabad city in India. *Energy Nexus* 5: 100044.
- Morando F., Heris M. P., Zulian G., Udías A., Mentaschi L., Chrysoulakis N., Parastatidis D., Maes J., 2022: Urban heat island mitigation by green infrastructure in European Functional Urban Areas. *Sustainable Cities and Society* 77: 103564.
- Oke, T. R., 1982: The energetic basis of the urban heat island. *Quarterly Journal of the Royal Meteorological Society*, 108(455): 1–24.

Oukawa Y. G., Krecl P., Targino C. A., 2022: Fine-scale modeling of the urban heat island: A comparison of multiple linear regression and random forest approaches. *Science of The Total Environment* 815: 152836.

Peng S., Piao S., Ciais P., Friedlingstein P., Oettle C., Breon F. -M., Nan H., Zhou L., Myneni R. B., 2011: Surface Urban Heat Island Across 419 Global Big Cities. *Environmental Science & Technology* 46(2): 696–703.

Rani M., Kumar P., Pandey P. C., Srivastava P. K., Chaudhary B. S., Tomar V., Mandal V. P., 2018: Multi-temporal NDVI and surface temperature analysis for Urban Heat Island inbuilt surrounding of sub-humid region: A case study of two geographical regions. *Remote Sensing Applications: Society and Environment* 10: 163–172.

Ritchie H. & Roser M., 2018: Urbanization (online) [cit. 2022.02.28.], dostupné z <<https://ourworldindata.org/urbanization>>.

RIZWAN A. M., DENNIS L. Y. C., LIU C., 2008: A review on the generation, determination and mitigation of Urban Heat Island. *Journal of Environmental Sciences*, 20(1), 120–128.

Schaaf C., Wang Z., 2015: MCD43A3 MODIS/Terra+Aqua BRDF/Albedo Daily L3 Global – 500m V006. NASA EOSDIS Land Processes DAAC.

Sobrino J. A., Oltra-Carrió R., Sòria G., Jiménez-Muñoz J. C., Franch B., Hidalgo, V., Mattar C., Julien Y., Cuenca J., Romaguera M., Gómez A. J., Miguel E., Bianchi R., & Paganini M., 2013: Evaluation of the surface urban heat island effect in the city of Madrid by thermal remote sensing. *International Journal of Remote Sensing* 34: 3177.

Stathopoulou M., Cartalis C., Andritsos A., 2005: Assessing the thermal environment of major cities in Greece. *International Conference “Passive and Low Energy Cooling for the Built Environment”*: 108-112.

Tan J. K. N., Belcher R. N., Tan H. T. W., Menz S., Schroeffer T., 2021: The urban heat island mitigation potential of vegetation depends on local surface type and shade. *Urban Forestry & Urban Greening* 62: 127128.

Tranmer M., Murphy J., Elliot M., Pampaka M., 2020: *Multiple Linear Regression (2nd Edition)*. Cathie Marsh Institute Working Paper, 57 s.

Trlica A., Hutyrá L. R., Schaaf C. L., Erb A., Wang J. A., 2017: Albedo, Land Cover, and Daytime Surface Temperature Variation Across an Urbanized Landscape. *Earth's Future* 5(11): 1084–1101.

U.S. Environmental Protection Agency, ©2008: What are Heat Islands? (online) [cit. 2022.03.02.], dostupné z <<https://www.epa.gov/heatislands/learn-about-heat-islands>>.

Vacek O., Kunt M., Čechová K., 2018: Městský tepelný ostrov. *NIKA* 39: 18-21.

Vermote F. E., Roger C.J., Ray P. J., 2015: MODIS Surface Reflectance User's Guide Collection 6. MODIS Land Surface Reflectance Science Computing Facility, 35 s.

Wan Z., 2013: Collection-6 MODIS Land Surface Temperature Products Users' Guide. ERI, University of California, Santa Barbara.

Wang Z., Meng Q., Allam M., Hu D., Zhang L., Menenti M., 2021: Environmental and anthropogenic drivers of surface urban heat island intensity: A case-study in the Yangtze River Delta, China. *Ecological Indicators* 128: 107845.

Weier J. & Herring D., 30.08.2000: Measuring Vegetation (NDVI & EVI) (online) [cit. 2022.03.01.], dostupné z <<https://earthobservatory.nasa.gov/>>.

WorldPop, ©2016: WorldPop Project (online) [cit.2022.03.12.], dostupné z <<https://hub.worldpop.org/portfolio/projects>>.

Xu L. Y., Xie X. D., Li S., 2013: Correlation analysis of the urban heat island effect and the spatial and temporal distribution of atmospheric particulates using TM images in Beijing. *Environmental Pollution* 178: 102–114.

Zhang, J., & Wang, Y., 2008: Study of the Relationships between the Spatial Extent of Surface Urban Heat Islands and Urban Characteristic Factors Based on Landsat ETM+ Data. *Sensors* 8(11): 7453–7468.

Zhou D., Zhao S., Liu S., Zhang L., Zhu C., 2014: Surface urban heat island in China's 32 major cities: Spatial patterns and drivers. *Remote Sensing of Environment* 152: 51-61.

10. Seznam příloh

Příloha č.1: Skript (a) – Měsíční průměry SUHII ve městech

Příloha č.2: Skript (b) – Průměrné hodnoty řídicích faktorů a průměrné SUHII ve městech

Příloha č.3: Skript (c) – Vytvoření vzorků pro regresní analýzu

Příloha č. 4: Absolutní standardizované regresní koeficienty

Příloha č. 1: Skript (a) – Měsíční průměry SUHI ve městech

```
////////// BACHELOR THESIS ////////////
////////// MADE BY KRYŠTOF VÁVRA ////////////

//// INPUTS //// ** YOU CAN CHANGE ANY OF THE INPUTS

// Cities, choose only from the available data
var CITIES = ["Amsterdam", "Berlin", "Budapest", "Madrid", "Stockholm", "London",
"Paris", "Prague", "Rome", "Warsaw"]; // "Amsterdam", "Berlin", "Budapest",
"Madrid", "Stockholm", "London", "Paris", "Prague", "Rome", "Warsaw"

// Years from 2000 to 2020
var YEAR_FROM = 2015;
var YEAR_TO = 2020;

// Months, you need to change months in both arrays
var MONTHS_TO_INCLUDE = [1,2,3,4,5,6,7,8,9,10,11,12]; // NUMBERS OF DESIRED MONTHS
var MONTHS_NAMES = ["JANUARY", "FEBRUARY", "MARCH", "APRIL", "MAY", "JUNE",
"JULY", "AUGUST", "SEPTEMBER", "OCTOBER", "NOVEMBER", "DECEMBER"]; // NAMES OF THE
MONTHS

// Days
var DAY_FROM = 1;
var DAY_TO = 31;

// Year of Corine Land Cover
var CLC_YEAR = 2018; // ONLY YEARS 2018, 2012, 2006, 2000, 1990 !

//// ARRAYS ////

// Arrays of average monthly values in urban areas
var SUHI_MONTHS_NIGHT = [];
var SUHI_MONTHS_DAY = [];
var NDVI_MONTHS = [];
var POPULATION_MONTHS = [];
var NDWI_MONTHS = [];
var LIGHT_MONTHS = [];
var ALBEDO_MONTHS = [];

// Arrays of yearly averages in urban areas
var SUHI_CITIES_NIGHT = [];
var SUHI_CITIES_DAY = [];
var NDVI_CITIES = [];
var SRTM_CITIES = [];
var POPULATION_CITIES = [];
var AREA_CITIES = [];
var NDWI_CITIES = [];
var LIGHT_CITIES = [];
var ALBEDO_CITIES = [];
var IMPERVIOUS_SURFACES_CITIES = [];

//// FUNCTIONS

// LST K to C
function LST_K_to_C (image) {
    return (image.multiply(0.02).subtract(273.15));
}

// NDVI scaled
function NDVI_SCALED (image) {
    return (image.multiply(0.0001));
}

//// GOING THROUGH ALL THE CITIES

for (var cities_number = 0; cities_number < CITIES.length; cities_number ++) {

    //// ROUTES
```



```

    var route_rural_border = "projects/uhi-thesis/assets/rural_borders/" +
CITIES[cities_number].toLowerCase() + "_rural_border";
    var route_urban_border = "projects/uhi-thesis/assets/urban_borders/" +
CITIES[cities_number].toLowerCase() + "_urban_border";
    var route_impervious_surfaces = "projects/uhi-
thesis/assets/impervious_surfaces/impervious_surfaces_" +
CITIES[cities_number].toLowerCase();

    /// IMPORTS

var LST_MODIS = ee.ImageCollection("MODIS/061/MOD11A1"), // LST MODIS
    SRTM = ee.Image("USGS/SRTMGL1_003"), // SRTM
    NDVI_MODIS = ee.ImageCollection("MODIS/006/MOD13A2"), // NDVI
    POPULATION = ee.ImageCollection("WorldPop/GP/100m/pop"), // POPULATION DATA
    NDWI_MODIS = ee.ImageCollection('MODIS/MOD09GA_006_NDWI'),
    LIGHT_RADIATION = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSLCFG"),
    ALBEDO_MODIS = ee.ImageCollection("MODIS/006/MCD43A3"),
    IMPERVIOUS_SURFACE = ee.Image(route_impervious_surfaces),
    CLC = ee.Image("COPERNICUS/CORINE/V20/100m/" + CLC_YEAR), // CLC
    URBAN_BORDER = ee.FeatureCollection(route_urban_border), // URBAN AREA OF
INTEREST
    RURAL_BORDER = ee.FeatureCollection(route_rural_border); // RURAL AREA OF
INTEREST

    /// GOING THROUGH EACH MONTH

for (var i = 0; i < MONTHS_TO_INCLUDE.length; i++) {

    var month = MONTHS_TO_INCLUDE[i];

    /// LST

    // MODIS LST mean by months
var LST_MONTHLY_MEAN = LST_MODIS
    .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
    .filter(ee.Filter.calendarRange(month, month, "month"))
    .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
    .mean();

    // MODIS LST night
var LST_NIGHT = LST_MONTHLY_MEAN
    .select("LST_Night_1km");

    // MODIS LST day
var LST_DAY = LST_MONTHLY_MEAN
    .select("LST_Day_1km");

    /// MODIS NDVI

    // MODIS NDVI mean by months
var NDVI_MONTHLY_MEAN = NDVI_MODIS
    .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
    .filter(ee.Filter.calendarRange(month, month, "month"))
    .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
    .mean();

    /// POPULATION DATA

    // Population mean only for each year
var POPULATION_MONTHLY_MEAN = POPULATION
    .filter(ee.Filter.calendarRange(01, 01, "day_of_month"))
    .filter(ee.Filter.calendarRange(01, 01, "month"))
    .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
    .mean()
    .cast({population: "double"});

    /// WATER PROPORTION

```

```

// NDWI mean by months
var NDWI_MONTHLY_MEAN = NDWI_MODIS
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// LIGHT RADIATION

// LIGHT mean by months
var LIGHT_MONTHLY_MEAN = LIGHT_RADIATION
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// ALBEDO

// Albedo mean by months
var ALBEDO_MONTHLY_MEAN = ALBEDO_MODIS
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// CLC URBAN

// CLC urban classes
var CLC_URBAN_RASTER = CLC
  .updateMask(CLC.gte(111).and(CLC.lte(124)));

// CLC urban raster to vector conversion
var CLC_URBAN = CLC_URBAN_RASTER
  .reduceToVectors({
    geometry: URBAN_BORDER
  });

//// LST URBAN

// LST MODIS urban night
var LST_URBAN_NIGHT = LST_NIGHT
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

// LST MODIS urban day
var LST_URBAN_DAY = LST_DAY
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

//// LST RURAL

// Maximum elevation in urban areas
var MAX_VAL_URBAN_ELEVATION = ee.Number(
  SRTM
  .reduceRegion({
    reducer: ee.Reducer.max(), // max value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  }).get("elevation")
);

// Minimum elevation in urban areas
var MIN_VAL_URBAN_ELEVATION = ee.Number(
  SRTM
  .reduceRegion({
    reducer: ee.Reducer.min(), // min value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  }).get("elevation")
);

```

```

);

/// CLC RURAL

// CLC rural classes
var CLC_RURAL_RASTER = CLC
  .updateMask(CLC.gte(211).and(CLC.lte(335)));

// CLC raster to vector conversion
var CLC_RURAL = CLC_RURAL_RASTER
  .reduceToVectors({
    geometry: RURAL_BORDER
  });

/// SRTM RURAL

// SRTM elevation max difference <> +/- 50 meters urban areas
var SRTM_RURAL_AREAS_RASTER = SRTM

.updateMask(SRTM.lte(MAX_VAL_URBAN_ELEVATION.add(30)).and(SRTM.gte(MIN_VAL_URBAN_E
LEVATION.subtract(30))));

// CLC raster to vector conversion
var SRTM_RURAL = SRTM_RURAL_AREAS_RASTER
  .reduceToVectors({
    geometry: RURAL_BORDER, scale: 1000
  });

// LST MODIS rural night
var LST_RURAL_NIGHT = LST_NIGHT
  .clip(RURAL_BORDER)
  .clip(CLC_RURAL)
  .clip(SRTM_RURAL);

// LST MODIS rural day
var LST_RURAL_DAY = LST_DAY
  .clip(RURAL_BORDER)
  .clip(CLC_RURAL)
  .clip(SRTM_RURAL);

/// NDVI URBAN

var NDVI_URBAN = NDVI_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// SRTM URBAN

var SRTM_URBAN = SRTM
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// POPULATION URBAN

var POPULATION_URBAN = POPULTAION_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// NDWI URBAN

var NDWI_URBAN = NDWI_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// LIGHT URBAN

var LIGHT_URBAN = LIGHT_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

```

```

/// ALBEDO URBAN

var ALBEDO_URBAN = ALBEDO_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// IMPERVIOUS SURFACE URBAN

var IMPERVIOUS_SURFACE_URBAN = IMPERVIOUS_SURFACE
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

//// AVERAGE VALUES ////

/// LST URBAN MEAN

// LST urban mean value night
var LST_URBAN_MEAN_NIGHT = ee.Number(
  LST_K_to_C(LST_URBAN_NIGHT)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  })).get("LST_Night_1km")
);

// LST urban mean value day
var LST_URBAN_MEAN_DAY = ee.Number(
  LST_K_to_C(LST_URBAN_DAY)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  })).get("LST_Day_1km")
);

/// LST RURAL MEAN

// LST rural mean value night
var LST_RURAL_MEAN_NIGHT = ee.Number(
  LST_K_to_C(LST_RURAL_NIGHT)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_RURAL, // polygon region
    scale: 1000 // 1000m
  })).get("LST_Night_1km")
);

// LST rural mean value day
var LST_RURAL_MEAN_DAY = ee.Number(
  LST_K_to_C(LST_RURAL_DAY)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_RURAL, // polygon region
    scale: 1000 // 1000m
  })).get("LST_Day_1km")
);

/// NDVI AND SRTM

// NDVI
var NDVI_URBAN_MEAN = ee.Number(
  NDVI_SCALED(NDVI_URBAN)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  })).get("NDVI")

```

```

);

// SRTM
var SRTM_URBAN_MEAN = ee.Number(
  SRTM_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  }).get("elevation")
);

// Population
var POPULATION_URBAN_SUM = ee.Number(
  POPULATION_URBAN
  .reduceRegion({
    reducer: ee.Reducer.sum(), // sum value
    geometry: CLC_URBAN, // polygon region
    scale: 100 // 100m
  }).get("population")
);

// NDWI
var NDWI_URBAN_MEAN = ee.Number(
  NDWI_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 463 // 463m
  }).get("NDWI")
);

// LIGHT
var LIGHT_URBAN_MEAN = ee.Number(
  LIGHT_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 463 // 463m
  }).get("avg_rad")
);

// ALBEDO
var ALBEDO_URBAN_MEAN = ee.Number(
  ALBEDO_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 500 // 500m
  }).get("Albedo_WSA_shortwave")
).multiply(0.001);

// IMPERVIOUS SURFACE
var IMPERVIOUS_SURFACE_URBAN_MEAN = ee.Number(
  IMPERVIOUS_SURFACE_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 100 // 100m
  }).get("b1")
);

/// TOTAL AREA OF URBAN AREAS square kilometers
var URBAN_AREA = CLC_URBAN.geometry().area(1).divide(1000000);

//// SUHI ////

// SUHI night

```

```

var SUHI_NIGHT = LST_URBAN_MEAN_NIGHT.subtract(LST_RURAL_MEAN_NIGHT);

// SUHI day
var SUHI_DAY = LST_URBAN_MEAN_DAY.subtract(LST_RURAL_MEAN_DAY);

/// WRITING TO ITS ARRAY

// SUHI
SUHI_MONTHS_NIGHT.push(SUHI_NIGHT);
SUHI_MONTHS_DAY.push(SUHI_DAY);

// NDVI
NDVI_MONTHS.push(NDVI_URBAN_MEAN);

// POPULATION
if (month === 1) {
  POPULATION_MONTHS.push(POPULATION_URBAN_SUM);
}

// NDWI
NDWI_MONTHS.push(NDWI_URBAN_MEAN);

// LIGHT RADIATION
LIGHT_MONTHS.push(LIGHT_URBAN_MEAN);

// ALBEDO RADIATION
ALBEDO_MONTHS.push(ALBEDO_URBAN_MEAN);

//// PRINTS AND CHARTS

if (i == MONTHS_TO_INCLUDE.length - 1) {

  /// AVERAGES OF MONTHLY AVERAGES = YEAR AVERAGES

  // NIGHT SUHI
  var SUHI_NIGHT_CITY_AVERAGE =
ee.Number(ee.List(SUHI_MONTHS_NIGHT).reduce(ee.Reducer.mean()));

  // DAY SUHI
  var SUHI_DAY_CITY_AVERAGE =
ee.Number(ee.List(SUHI_MONTHS_DAY).reduce(ee.Reducer.mean()));

  // NDVI
  var NDVI_CITY_AVERAGE =
ee.Number(ee.List(NDVI_MONTHS).reduce(ee.Reducer.mean()));

  // POPULATION
  var POPULATION_CITY_AVERAGE =
ee.Number(ee.List(POPULATION_MONTHS).reduce(ee.Reducer.mean()));

  // NDWI
  var NDWI_CITY_AVERAGE =
ee.Number(ee.List(NDWI_MONTHS).reduce(ee.Reducer.mean()));

  // LIGHT
  var LIGHT_CITY_AVERAGE =
ee.Number(ee.List(LIGHT_MONTHS).reduce(ee.Reducer.mean()));

  // ALBEDO
  var ALBEDO_CITY_AVERAGE =
ee.Number(ee.List(ALBEDO_MONTHS).reduce(ee.Reducer.mean()));

  // WRITING TO ITS ARRAY
  SUHI_CITIES_NIGHT.push(SUHI_NIGHT_CITY_AVERAGE);
  SUHI_CITIES_DAY.push(SUHI_DAY_CITY_AVERAGE);
  NDVI_CITIES.push(NDVI_CITY_AVERAGE);
  SRTM_CITIES.push(SRTM_URBAN_MEAN);
  POPULATION_CITIES.push(POPULATION_CITY_AVERAGE);
  AREA_CITIES.push(URBAN_AREA);
}

```

```

NDWI_CITIES.push(NDWI_CITY_AVERAGE);
LIGHT_CITIES.push(LIGHT_CITY_AVERAGE);
ALBEDO_CITIES.push(ALBEDO_CITY_AVERAGE);
IMPERVIOUS_SURFACES_CITIES.push(IMPERVIOUS_SURFACE_URBAN_MEAN);

// Emptying array
var SUHI_MONTHS_NIGHT = [];
var SUHI_MONTHS_DAY = [];
var NDVI_MONTHS = [];
var POPULATION_MONTHS = [];
var NDWI_MONTHS = [];
var LIGHT_MONTHS = [];
var ALBEDO_MONTHS = [];

// Creating a chart for total averages in all cities when the array is
completed
if (cities_number == CITIES.length - 1) {

    /// NIGHT SUHI

    // List
    var SUHI_CITIES_ARRAY_NIGHT = ee.List(SUHI_CITIES_NIGHT);

    // Define the chart and print it to the console.
    var CHART_CITIES_NIGHT = ui.Chart.array.values({array:
SUHI_CITIES_ARRAY_NIGHT, axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "Average NIGHT SUHI intensity per city, " + YEAR_FROM + " to "
+ YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE SUHI INTENSITY",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['red'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_NIGHT);

    /// DAY SUHI

    // List
    var SUHI_CITIES_ARRAY_DAY = ee.List(SUHI_CITIES_DAY);

    // Define the chart and print it to the console.
    var CHART_CITIES_SUHI_DAY = ui.Chart.array.values({array:
SUHI_CITIES_ARRAY_DAY, axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "Average DAY SUHI intensity per city, " + YEAR_FROM + " to " +
YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE SUHI INTENSITY",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['red'],
            lineSize: 5,

```

```

        pointSize: 10,
        legend: {position: 'none'}
    });
    print(CHART_CITIES_SUHI_DAY);

    /// NDVI

    // List
    var NDVI_CITIES_ARRAY = ee.List(NDVI_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_NDVI = ui.Chart.array.values({array: NDVI_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "Average NDVI per city, " + YEAR_FROM + " to " + YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE NDVI",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['green'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_NDVI);

    /// SRTM

    // List
    var SRTM_CITIES_ARRAY = ee.List(SRTM_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_SRTM = ui.Chart.array.values({array: SRTM_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "Average elevation of urban areas per city, " + YEAR_FROM + "
to " + YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE ELEVATION (m)",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['brown'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_SRTM);

    /// AREA

    // List
    var AREA_CITIES_ARRAY = ee.List(AREA_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_AREA = ui.Chart.array.values({array: AREA_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")

```



```

.setOptions({
  title: "Calculated total urban area in cities",
  titleTextStyle: {italic: false, bold: true, fontSize: 15},
  hAxis: {
    title: "CITIES",
    titleTextStyle: {italic: false, bold: true}
  },
  vAxis: {
    title: "AREA (SQUARE KM)",
    titleTextStyle: {italic: false, bold: true}
  },
  colors: ['orange'],
  lineSize: 5,
  pointSize: 10,
  legend: {position: 'none'}
});
print(CHART_CITIES_AREA);

/// POPULATION

// List
var POPULATION_CITIES_ARRAY = ee.List(POPULATION_CITIES);

// Define the chart and print it to the console.
var CHART_CITIES_POPULATION = ui.Chart.array.values({array:
POPULATION_CITIES_ARRAY, axis: 0, xLabels: CITIES})
.setChartType("ColumnChart")
.setOptions({
  title: "Total population of urban areas per city, " + YEAR_FROM + " to
" + YEAR_TO,
  titleTextStyle: {italic: false, bold: true, fontSize: 15},
  hAxis: {
    title: "CITIES",
    titleTextStyle: {italic: false, bold: true}
  },
  vAxis: {
    title: "POPULATION",
    titleTextStyle: {italic: false, bold: true}
  },
  colors: ['purple'],
  lineSize: 5,
  pointSize: 10,
  legend: {position: 'none'}
});
print(CHART_CITIES_POPULATION);

/// NDWI

// List
var NDWI_CITIES_ARRAY = ee.List(NDWI_CITIES);

// Define the chart and print it to the console.
var CHART_CITIES_NDWI = ui.Chart.array.values({array: NDWI_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
.setChartType("ColumnChart")
.setOptions({
  title: "AVERAGE NDWI of urban areas per city, " + YEAR_FROM + " to " +
YEAR_TO,
  titleTextStyle: {italic: false, bold: true, fontSize: 15},
  hAxis: {
    title: "CITIES",
    titleTextStyle: {italic: false, bold: true}
  },
  vAxis: {
    title: "NDWI",
    titleTextStyle: {italic: false, bold: true}
  },
  colors: ['blue'],
  lineSize: 5,

```

```

        pointSize: 10,
        legend: {position: 'none'}
    });
    print(CHART_CITIES_NDWI);

    /// LIGHT

    // List
    var LIGHT_CITIES_ARRAY = ee.List(LIGHT_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_LIGHT = ui.Chart.array.values({array: LIGHT_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "AVERAGE RADIANCE of urban areas per city, " + YEAR_FROM + " to
" + YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE DNB RADIANCE VALUES (nanoWatts/cm2/sr)",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['yellow'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_LIGHT);

    /// ALBEDO

    // List
    var ALBEDO_CITIES_ARRAY = ee.List(ALBEDO_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_ALBEDO = ui.Chart.array.values({array:
ALBEDO_CITIES_ARRAY, axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "AVERAGE White sky ALBEDO for SHORTWAVE BROADBAND of urban
areas per city, " + YEAR_FROM + " to " + YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "WHITE-SKY ALBEDO ",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['grey'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_ALBEDO);

    // List
    var IMPERVIOUS_SURFACES_CITIES_ARRAY =
ee.List(IMPERVIOUS_SURFACES_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_IMPERVIOUS_SURFACES = ui.Chart.array.values({array:
IMPERVIOUS_SURFACES_CITIES_ARRAY, axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")

```

```

        .setOptions({
            title: "AVERAGE IMPERVIOUS SURFACES of urban areas per city",
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "IMPERVIOUS SURFACES (%)",
                titleTextStyle: {italic: false, bold: true},
                viewWindow: {min: 0, max: 100}
            },
            colors: ['black'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
        print(CHART_CITIES_IMPERVIOUS_SURFACES);
    }

}

}

}

//// VIEW OF EUROPE
Map.setCenter(14.869854282951186, 48.69447525861791, 5);

```

Příloha č. 2: Skript (b) – Průměrné hodnoty řídicích faktorů a průměrné SUHI ve městech

```
////////// BACHELOR THESIS ////////////  
////////// MADE BY KRYŠTOF VÁVRA ////////////  
  
//// INPUTS //// ** YOU CAN CHANGE ANY OF THE INPUTS  
  
// Cities, choose only from the available data  
var CITIES = ["Amsterdam", "Berlin", "Budapest", "Madrid", "Stockholm", "London",  
"Paris", "Prague", "Rome", "Warsaw"]; // "Amsterdam", "Berlin", "Budapest",  
"Madrid", "Stockholm", "London", "Paris", "Prague", "Rome", "Warsaw"  
  
// Years from 2000 to 2020  
var YEAR_FROM = 2015;  
var YEAR_TO = 2020;  
  
// Months, you need to change months in both arrays  
var MONTHS_TO_INCLUDE = [1,2,3,4,5,6,7,8,9,10,11,12]; // NUMBERS OF DESIRED MONTHS  
var MONTHS_NAMES = ["JANUARY", "FEBRUARY", "MARCH", "APRIL", "MAY", "JUNE",  
"JULY", "AUGUST", "SEPTEMBER", "OCTOBER", "NOVEMBER", "DECEMBER"]; // NAMES OF THE  
MONTHS  
  
// Days  
var DAY_FROM = 1;  
var DAY_TO = 31;  
  
// Year of Corine Land Cover  
var CLC_YEAR = 2018; // ONLY YEARS 2018, 2012, 2006, 2000, 1990 !  
  
//// ARRAYS ////  
  
// Arrays of average monthly values in urban areas  
var SUHI_MONTHS_NIGHT = [];  
var SUHI_MONTHS_DAY = [];  
var NDVI_MONTHS = [];  
var POPULATION_MONTHS = [];  
var NDWI_MONTHS = [];  
var LIGHT_MONTHS = [];  
var ALBEDO_MONTHS = [];  
  
// Arrays of yearly averages in urban areas  
var SUHI_CITIES_NIGHT = [];  
var SUHI_CITIES_DAY = [];  
var NDVI_CITIES = [];  
var SRTM_CITIES = [];  
var POPULATION_CITIES = [];  
var AREA_CITIES = [];  
var NDWI_CITIES = [];  
var LIGHT_CITIES = [];  
var ALBEDO_CITIES = [];  
var IMPERVIOUS_SURFACES_CITIES = [];  
  
//// FUNCTIONS  
  
// LST K to C  
function LST_K_to_C (image) {  
    return (image.multiply(0.02).subtract(273.15));  
}  
  
// NDVI scaled  
function NDVI_SCALED (image) {  
    return (image.multiply(0.0001));  
}  
  
//// GOING THROUGH ALL THE CITIES  
  
for (var cities_number = 0; cities_number < CITIES.length; cities_number++) {  
  
    //// ROUTES
```

```

    var route_rural_border = "projects/uhi-thesis/assets/rural_borders/" +
CITIES[cities_number].toLowerCase() + "_rural_border";
    var route_urban_border = "projects/uhi-thesis/assets/urban_borders/" +
CITIES[cities_number].toLowerCase() + "_urban_border";
    var route_impervious_surfaces = "projects/uhi-
thesis/assets/impervious_surfaces/impervious_surfaces_" +
CITIES[cities_number].toLowerCase();

    /// IMPORTS

var LST_MODIS = ee.ImageCollection("MODIS/061/MOD11A1"), // LST MODIS
    SRTM = ee.Image("USGS/SRTMGL1_003"), // SRTM
    NDVI_MODIS = ee.ImageCollection("MODIS/006/MOD13A2"), // NDVI
    POPULATION = ee.ImageCollection("WorldPop/GP/100m/pop"), // POPULATION DATA
    NDWI_MODIS = ee.ImageCollection('MODIS/MOD09GA_006_NDWI'),
    LIGHT_RADIATION = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSLCFG"),
    ALBEDO_MODIS = ee.ImageCollection("MODIS/006/MCD43A3"),
    IMPERVIOUS_SURFACE = ee.Image(route_impervious_surfaces),
    CLC = ee.Image("COPERNICUS/CORINE/V20/100m/" + CLC_YEAR), // CLC
    URBAN_BORDER = ee.FeatureCollection(route_urban_border), // URBAN AREA OF
INTEREST
    RURAL_BORDER = ee.FeatureCollection(route_rural_border); // RURAL AREA OF
INTEREST

    /// GOING THROUGH EACH MONTH

for (var i = 0; i < MONTHS_TO_INCLUDE.length; i++) {

    var month = MONTHS_TO_INCLUDE[i];

    /// LST

    // MODIS LST mean by months
var LST_MONTHLY_MEAN = LST_MODIS
    .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
    .filter(ee.Filter.calendarRange(month, month, "month"))
    .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
    .mean();

    // MODIS LST night
var LST_NIGHT = LST_MONTHLY_MEAN
    .select("LST_Night_1km");

    // MODIS LST day
var LST_DAY = LST_MONTHLY_MEAN
    .select("LST_Day_1km");

    /// MODIS NDVI

    // MODIS NDVI mean by months
var NDVI_MONTHLY_MEAN = NDVI_MODIS
    .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
    .filter(ee.Filter.calendarRange(month, month, "month"))
    .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
    .mean();

    /// POPULATION DATA

    // Population mean only for each year
var POPULATION_MONTHLY_MEAN = POPULATION
    .filter(ee.Filter.calendarRange(01, 01, "day_of_month"))
    .filter(ee.Filter.calendarRange(01, 01, "month"))
    .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
    .mean()
    .cast({population: "double"});

    /// WATER PROPORTION

```

```

// NDWI mean by months
var NDWI_MONTHLY_MEAN = NDWI_MODIS
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// LIGHT RADIATION

// LIGHT mean by months
var LIGHT_MONTHLY_MEAN = LIGHT_RADIATION
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// ALBEDO

// Albedo mean by months
var ALBEDO_MONTHLY_MEAN = ALBEDO_MODIS
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// CLC URBAN

// CLC urban classes
var CLC_URBAN_RASTER = CLC
  .updateMask(CLC.gte(111).and(CLC.lte(124)));

// CLC urban raster to vector conversion
var CLC_URBAN = CLC_URBAN_RASTER
  .reduceToVectors({
    geometry: URBAN_BORDER
  });

//// LST URBAN

// LST MODIS urban night
var LST_URBAN_NIGHT = LST_NIGHT
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

// LST MODIS urban day
var LST_URBAN_DAY = LST_DAY
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

//// LST RURAL

// Maximum elevation in urban areas
var MAX_VAL_URBAN_ELEVATION = ee.Number(
  SRTM
  .reduceRegion({
    reducer: ee.Reducer.max(), // max value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  }).get("elevation")
);

// Minimum elevation in urban areas
var MIN_VAL_URBAN_ELEVATION = ee.Number(
  SRTM
  .reduceRegion({
    reducer: ee.Reducer.min(), // min value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  }).get("elevation")
);

```

```

);

/// CLC RURAL

// CLC rural classes
var CLC_RURAL_RASTER = CLC
  .updateMask(CLC.gte(211).and(CLC.lte(335)));

// CLC raster to vector conversion
var CLC_RURAL = CLC_RURAL_RASTER
  .reduceToVectors(
    {geometry: RURAL_BORDER}
  );

/// SRTM RURAL

// SRTM elevation max difference <> +/- 50 meters urban areas
var SRTM_RURAL_AREAS_RASTER = SRTM

.updateMask(SRTM.lte(MAX_VAL_URBAN_ELEVATION.add(30)).and(SRTM.gte(MIN_VAL_URBAN_E
LEVATION.subtract(30))));

// CLC raster to vector conversion
var SRTM_RURAL = SRTM_RURAL_AREAS_RASTER
  .reduceToVectors(
    {geometry: RURAL_BORDER, scale: 1000}
  );

// LST MODIS rural night
var LST_RURAL_NIGHT = LST_NIGHT
  .clip(RURAL_BORDER)
  .clip(CLC_RURAL)
  .clip(SRTM_RURAL);

// LST MODIS rural day
var LST_RURAL_DAY = LST_DAY
  .clip(RURAL_BORDER)
  .clip(CLC_RURAL)
  .clip(SRTM_RURAL);

/// NDVI URBAN

var NDVI_URBAN = NDVI_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// SRTM URBAN

var SRTM_URBAN = SRTM
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// POPULATION URBAN

var POPULATION_URBAN = POPULTAION_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// NDWI URBAN

var NDWI_URBAN = NDWI_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// LIGHT URBAN

var LIGHT_URBAN = LIGHT_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

```

```

/// ALBEDO URBAN

var ALBEDO_URBAN = ALBEDO_MONTHLY_MEAN
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

/// IMPERVIOUS SURFACE URBAN

var IMPERVIOUS_SURFACE_URBAN = IMPERVIOUS_SURFACE
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

//// AVERAGE VALUES ////

/// LST URBAN MEAN

// LST urban mean value night
var LST_URBAN_MEAN_NIGHT = ee.Number(
  LST_K_to_C(LST_URBAN_NIGHT)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  })).get("LST_Night_1km")
);

// LST urban mean value day
var LST_URBAN_MEAN_DAY = ee.Number(
  LST_K_to_C(LST_URBAN_DAY)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  })).get("LST_Day_1km")
);

/// LST RURAL MEAN

// LST rural mean value night
var LST_RURAL_MEAN_NIGHT = ee.Number(
  LST_K_to_C(LST_RURAL_NIGHT)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_RURAL, // polygon region
    scale: 1000 // 1000m
  })).get("LST_Night_1km")
);

// LST rural mean value day
var LST_RURAL_MEAN_DAY = ee.Number(
  LST_K_to_C(LST_RURAL_DAY)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_RURAL, // polygon region
    scale: 1000 // 1000m
  })).get("LST_Day_1km")
);

/// NDVI AND SRTM

// NDVI
var NDVI_URBAN_MEAN = ee.Number(
  NDVI_SCALED(NDVI_URBAN)
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  })).get("NDVI")

```



```

);

// SRTM
var SRTM_URBAN_MEAN = ee.Number(
  SRTM_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  }).get("elevation")
);

// Population
var POPULATION_URBAN_SUM = ee.Number(
  POPULATION_URBAN
  .reduceRegion({
    reducer: ee.Reducer.sum(), // sum value
    geometry: CLC_URBAN, // polygon region
    scale: 100 // 100m
  }).get("population")
);

// NDWI
var NDWI_URBAN_MEAN = ee.Number(
  NDWI_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 463 // 463m
  }).get("NDWI")
);

// LIGHT
var LIGHT_URBAN_MEAN = ee.Number(
  LIGHT_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 463 // 463m
  }).get("avg_rad")
);

// ALBEDO
var ALBEDO_URBAN_MEAN = ee.Number(
  ALBEDO_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 500 // 500m
  }).get("Albedo_WSA_shortwave")
).multiply(0.001);

// IMPERVIOUS SURFACE
var IMPERVIOUS_SURFACE_URBAN_MEAN = ee.Number(
  IMPERVIOUS_SURFACE_URBAN
  .reduceRegion({
    reducer: ee.Reducer.mean(), // mean value
    geometry: CLC_URBAN, // polygon region
    scale: 100 // 100m
  }).get("b1")
);

/// TOTAL AREA OF URBAN AREAS square kilometers
var URBAN_AREA = CLC_URBAN.geometry().area(1).divide(1000000);

//// SUHI ////

// SUHI night

```

```

var SUHI_NIGHT = LST_URBAN_MEAN_NIGHT.subtract(LST_RURAL_MEAN_NIGHT);

// SUHI day
var SUHI_DAY = LST_URBAN_MEAN_DAY.subtract(LST_RURAL_MEAN_DAY);

/// WRITING TO ITS ARRAY

// SUHI
SUHI_MONTHS_NIGHT.push(SUHI_NIGHT);
SUHI_MONTHS_DAY.push(SUHI_DAY);

// NDVI
NDVI_MONTHS.push(NDVI_URBAN_MEAN);

// POPULATION
if (month === 1) {
  POPULATION_MONTHS.push(POPULATION_URBAN_SUM);
}

// NDWI
NDWI_MONTHS.push(NDWI_URBAN_MEAN);

// LIGHT RADIATION
LIGHT_MONTHS.push(LIGHT_URBAN_MEAN);

// ALBEDO RADIATION
ALBEDO_MONTHS.push(ALBEDO_URBAN_MEAN);

//// PRINTS AND CHARTS

if (i == MONTHS_TO_INCLUDE.length - 1) {

  /// AVERAGES OF MONTHLY AVERAGES = YEAR AVERAGES

  // NIGHT SUHI
  var SUHI_NIGHT_CITY_AVERAGE =
ee.Number(ee.List(SUHI_MONTHS_NIGHT).reduce(ee.Reducer.mean()));

  // DAY SUHI
  var SUHI_DAY_CITY_AVERAGE =
ee.Number(ee.List(SUHI_MONTHS_DAY).reduce(ee.Reducer.mean()));

  // NDVI
  var NDVI_CITY_AVERAGE =
ee.Number(ee.List(NDVI_MONTHS).reduce(ee.Reducer.mean()));

  // POPULATION
  var POPULATION_CITY_AVERAGE =
ee.Number(ee.List(POPULATION_MONTHS).reduce(ee.Reducer.mean()));

  // NDWI
  var NDWI_CITY_AVERAGE =
ee.Number(ee.List(NDWI_MONTHS).reduce(ee.Reducer.mean()));

  // LIGHT
  var LIGHT_CITY_AVERAGE =
ee.Number(ee.List(LIGHT_MONTHS).reduce(ee.Reducer.mean()));

  // ALBEDO
  var ALBEDO_CITY_AVERAGE =
ee.Number(ee.List(ALBEDO_MONTHS).reduce(ee.Reducer.mean()));

  // WRITING TO ITS ARRAY
  SUHI_CITIES_NIGHT.push(SUHI_NIGHT_CITY_AVERAGE);
  SUHI_CITIES_DAY.push(SUHI_DAY_CITY_AVERAGE);
  NDVI_CITIES.push(NDVI_CITY_AVERAGE);
  SRTM_CITIES.push(SRTM_URBAN_MEAN);
  POPULATION_CITIES.push(POPULATION_CITY_AVERAGE);
  AREA_CITIES.push(URBAN_AREA);
}

```

```

NDWI_CITIES.push(NDWI_CITY_AVERAGE);
LIGHT_CITIES.push(LIGHT_CITY_AVERAGE);
ALBEDO_CITIES.push(ALBEDO_CITY_AVERAGE);
IMPERVIOUS_SURFACES_CITIES.push(IMPERVIOUS_SURFACE_URBAN_MEAN);

// Emptying array
var SUHI_MONTHS_NIGHT = [];
var SUHI_MONTHS_DAY = [];
var NDVI_MONTHS = [];
var POPULATION_MONTHS = [];
var NDWI_MONTHS = [];
var LIGHT_MONTHS = [];
var ALBEDO_MONTHS = [];

// Creating a chart for total averages in all cities when the array is
completed
if (cities_number == CITIES.length - 1) {

    /// NIGHT SUHI

    // List
    var SUHI_CITIES_ARRAY_NIGHT = ee.List(SUHI_CITIES_NIGHT);

    // Define the chart and print it to the console.
    var CHART_CITIES_NIGHT = ui.Chart.array.values({array:
SUHI_CITIES_ARRAY_NIGHT, axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "Average NIGHT SUHI intensity per city, " + YEAR_FROM + " to "
+ YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE SUHI INTENSITY",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['red'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_NIGHT);

    /// DAY SUHI

    // List
    var SUHI_CITIES_ARRAY_DAY = ee.List(SUHI_CITIES_DAY);

    // Define the chart and print it to the console.
    var CHART_CITIES_SUHI_DAY = ui.Chart.array.values({array:
SUHI_CITIES_ARRAY_DAY, axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "Average DAY SUHI intensity per city, " + YEAR_FROM + " to " +
YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE SUHI INTENSITY",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['red'],
            lineSize: 5,

```

```

        pointSize: 10,
        legend: {position: 'none'}
    });
    print(CHART_CITIES_SUHI_DAY);

    /// NDVI

    // List
    var NDVI_CITIES_ARRAY = ee.List(NDVI_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_NDVI = ui.Chart.array.values({array: NDVI_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "Average NDVI per city, " + YEAR_FROM + " to " + YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE NDVI",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['green'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_NDVI);

    /// SRTM

    // List
    var SRTM_CITIES_ARRAY = ee.List(SRTM_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_SRTM = ui.Chart.array.values({array: SRTM_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "Average elevation of urban areas per city, " + YEAR_FROM + "
to " + YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE ELEVATION (m)",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['brown'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_SRTM);

    /// AREA

    // List
    var AREA_CITIES_ARRAY = ee.List(AREA_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_AREA = ui.Chart.array.values({array: AREA_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")

```

```

.setOptions({
  title: "Calculated total urban area in cities",
  titleTextStyle: {italic: false, bold: true, fontSize: 15},
  hAxis: {
    title: "CITIES",
    titleTextStyle: {italic: false, bold: true}
  },
  vAxis: {
    title: "AREA (SQUARE KM)",
    titleTextStyle: {italic: false, bold: true}
  },
  colors: ['orange'],
  lineSize: 5,
  pointSize: 10,
  legend: {position: 'none'}
});
print(CHART_CITIES_AREA);

/// POPULATION

// List
var POPULATION_CITIES_ARRAY = ee.List(POPULATION_CITIES);

// Define the chart and print it to the console.
var CHART_CITIES_POPULATION = ui.Chart.array.values({array:
POPULATION_CITIES_ARRAY, axis: 0, xLabels: CITIES})
.setChartType("ColumnChart")
.setOptions({
  title: "Total population of urban areas per city, " + YEAR_FROM + " to
" + YEAR_TO,
  titleTextStyle: {italic: false, bold: true, fontSize: 15},
  hAxis: {
    title: "CITIES",
    titleTextStyle: {italic: false, bold: true}
  },
  vAxis: {
    title: "POPULATION",
    titleTextStyle: {italic: false, bold: true}
  },
  colors: ['purple'],
  lineSize: 5,
  pointSize: 10,
  legend: {position: 'none'}
});
print(CHART_CITIES_POPULATION);

/// NDWI

// List
var NDWI_CITIES_ARRAY = ee.List(NDWI_CITIES);

// Define the chart and print it to the console.
var CHART_CITIES_NDWI = ui.Chart.array.values({array: NDWI_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
.setChartType("ColumnChart")
.setOptions({
  title: "AVERAGE NDWI of urban areas per city, " + YEAR_FROM + " to " +
YEAR_TO,
  titleTextStyle: {italic: false, bold: true, fontSize: 15},
  hAxis: {
    title: "CITIES",
    titleTextStyle: {italic: false, bold: true}
  },
  vAxis: {
    title: "NDWI",
    titleTextStyle: {italic: false, bold: true}
  },
  colors: ['blue'],
  lineSize: 5,

```

```

        pointSize: 10,
        legend: {position: 'none'}
    });
    print(CHART_CITIES_NDWI);

    /// LIGHT

    // List
    var LIGHT_CITIES_ARRAY = ee.List(LIGHT_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_LIGHT = ui.Chart.array.values({array: LIGHT_CITIES_ARRAY,
axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "AVERAGE RADIANCE of urban areas per city, " + YEAR_FROM + " to
" + YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "AVERAGE DNB RADIANCE VALUES (nanoWatts/cm2/sr)",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['yellow'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_LIGHT);

    /// ALBEDO

    // List
    var ALBEDO_CITIES_ARRAY = ee.List(ALBEDO_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_ALBEDO = ui.Chart.array.values({array:
ALBEDO_CITIES_ARRAY, axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")
        .setOptions({
            title: "AVERAGE White sky ALBEDO for SHORTWAVE BROADBAND of urban
areas per city, " + YEAR_FROM + " to " + YEAR_TO,
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "WHITE-SKY ALBEDO ",
                titleTextStyle: {italic: false, bold: true}
            },
            colors: ['grey'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
    print(CHART_CITIES_ALBEDO);

    // List
    var IMPERVIOUS_SURFACES_CITIES_ARRAY =
ee.List(IMPERVIOUS_SURFACES_CITIES);

    // Define the chart and print it to the console.
    var CHART_CITIES_IMPERVIOUS_SURFACES = ui.Chart.array.values({array:
IMPERVIOUS_SURFACES_CITIES_ARRAY, axis: 0, xLabels: CITIES})
        .setChartType("ColumnChart")

```

```

        .setOptions({
            title: "AVERAGE IMPERVIOUS SURFACES of urban areas per city",
            titleTextStyle: {italic: false, bold: true, fontSize: 15},
            hAxis: {
                title: "CITIES",
                titleTextStyle: {italic: false, bold: true}
            },
            vAxis: {
                title: "IMPERVIOUS SURFACES (%)",
                titleTextStyle: {italic: false, bold: true},
                viewWindow: {min: 0, max: 100}
            },
            colors: ['black'],
            lineSize: 5,
            pointSize: 10,
            legend: {position: 'none'}
        });
        print(CHART_CITIES_IMPERVIOUS_SURFACES);
    }

}

}

}

//// VIEW OF EUROPE
Map.setCenter(14.869854282951186, 48.69447525861791, 5);

```

Příloha č. 3: Skript (c) – Vytvoření vzorků pro regresní analýzu

```
////////// BACHELOR THESIS ////////////
////////// MADE BY KRYŠTOF VÁVRA ////////////

//// INPUTS //// ** YOU CAN CHANGE ANY OF THE INPUTS

// Cities, choose only from the available data
var CITIES = [ "Prague"]; // Maybe its better to analyze each city alone and then
compare the results? ["Amsterdam", "Berlin", "Budapest", "Madrid", "Stockholm",
"London", "Paris", "Prague", "Rome", "Warsaw"]; // "Amsterdam", "Berlin",
"Budapest", "Madrid", "Stockholm", "London", "Paris", "Prague", "Rome", "Warsaw"

// Years from 2000 to 2020
var YEAR_FROM = 2015;
var YEAR_TO = 2020;

// Months
var MONTHS_TO_INCLUDE = [1,2,3,4,5,6,7,8,9,10,11,12]; // DESIRED MONTHS
1,2,3,4,5,6,7,8,9,10,11,12

// Days
var DAY_FROM = 1;
var DAY_TO = 31;

// Year of Corine Land Cover
var CLC_YEAR = 2018; // ONLY YEARS 2018, 2012, 2006, 2000, 1990 !

//// INPUTS ////

//// ARRAYS ////

// Arrays of average monthly values in urban areas
var SUHI_MONTHS_NIGHT = [];
var SUHI_MONTHS_DAY = [];
var NDVI_MONTHS = [];
var POPULATION_MONTHS = [];
var NDWI_MONTHS = [];
var LIGHT_MONTHS = [];
var ALBEDO_MONTHS = [];
var IMPERVIOUS_SURFACES_CITIES = [];

var URBAN_AREAS_GEOMETRIES = [];

//// FUNCTIONS

// LST K to C
function LST_K_to_C (image) {
    return (image.multiply(0.02).subtract(273.15));
}

// NDVI scaled
function NDVI_SCALED (image) {
    return (image.multiply(0.0001));
}

// NDVI scaled
function ALBEDO_SCALED (image) {
    return (image.multiply(0.001));
}

// Sampling points from the images
function point_sampling (bands, region, samples) {
    return (
        bands
        .sample({
            region: region,
```



```

        scale: 100,
        numPixels: samples,
        geometries: true
    });
}

function join_urban_areas (list_areas) {
    var urban_areas = list_areas[0];
    if (list_areas.length > 1) {
        for (var i = 1; i < list_areas.length; i++) {
            urban_areas = urban_areas.merge(list_areas[i])
            if (i == list_areas.length - 1) {
                return (urban_areas);
            }
        }
    } else {
        return(urban_areas)
    }
}

//// GOING THROUGH ALL THE CITIES

for (var cities_number = 0; cities_number < CITIES.length; cities_number++) {

    //// ROUTES

    var route_rural_border = "projects/uhi-thesis/assets/rural_borders/" +
    CITIES[cities_number].toLowerCase() + "_rural_border";
    var route_urban_border = "projects/uhi-thesis/assets/urban_borders/" +
    CITIES[cities_number].toLowerCase() + "_urban_border";
    var route_impervious_surfaces = "projects/uhi-
thesis/assets/impervious_surfaces/impervious_surfaces_" +
    CITIES[cities_number].toLowerCase();

    //// IMPORTS

    var LST_MODIS = ee.ImageCollection("MODIS/061/MOD11A1"), // LST MODIS
    SRTM = ee.Image("USGS/SRTMGL1_003"), // SRTM
    NDVI_MODIS = ee.ImageCollection("MODIS/006/MOD13A2"), // NDVI
    POPULATION = ee.ImageCollection("WorldPop/GP/100m/pop"), // POPULATION DATA
    NDWI_MODIS = ee.ImageCollection('MODIS/MOD09GA_006_NDWI'),
    LIGHT_RADIATION = ee.ImageCollection("NOAA/VIIRS/DNB/MONTHLY_V1/VCMSLCFG"),
    ALBEDO_MODIS = ee.ImageCollection("MODIS/006/MCD43A3"),
    IMPERVIOUS_SURFACE = ee.Image(route_impervious_surfaces),
    CLC = ee.Image("COPERNICUS/CORINE/V20/100m/" + CLC_YEAR), // CLC
    URBAN_BORDER = ee.FeatureCollection(route_urban_border), // URBAN AREA OF
INTEREST
    RURAL_BORDER = ee.FeatureCollection(route_rural_border); // RURAL AREA OF
INTEREST

    //// GOING THROUGH EACH MONTH

    for (var i = 0; i < MONTHS_TO_INCLUDE.length; i++) {

        var month = MONTHS_TO_INCLUDE[i];

        /// LST

        // MODIS LST mean by months
        var LST_MONTHLY_MEAN = LST_MODIS
            .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
            .filter(ee.Filter.calendarRange(month, month, "month"))
            .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
            .mean();

        // MODIS LST night
        var LST_NIGHT = LST_MONTHLY_MEAN
            .select("LST_Night_1km");
    }
}

```

```

// MODIS LST day
var LST_DAY = LST_MONTHLY_MEAN
  .select("LST_Day_1km");

/// MODIS NDVI

// MODIS NDVI mean by months
var NDVI_MONTHLY_MEAN = NDVI_MODIS
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// POPULATION DATA

// Population mean only for each year
var POPULATION_MONTHLY_MEAN = POPULATION
  .filter(ee.Filter.calendarRange(01, 01, "day_of_month"))
  .filter(ee.Filter.calendarRange(01, 01, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// WATER PROPORTION

// NDWI mean by months
var NDWI_MONTHLY_MEAN = NDWI_MODIS
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// LIGHT RADIATION

// LIGHT mean by months
var LIGHT_MONTHLY_MEAN = LIGHT_RADIATION
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// ALBEDO

// Albedo mean by months
var ALBEDO_MONTHLY_MEAN = ALBEDO_MODIS
  .filter(ee.Filter.calendarRange(DAY_FROM, DAY_TO, "day_of_month"))
  .filter(ee.Filter.calendarRange(month, month, "month"))
  .filter(ee.Filter.calendarRange(YEAR_FROM, YEAR_TO, "year"))
  .mean();

/// CLC URBAN

// CLC urban classes
var CLC_URBAN_RASTER = CLC
  .updateMask(CLC.gte(111).and(CLC.lte(124)));

// CLC urban raster to vector conversion
var CLC_URBAN = CLC_URBAN_RASTER
  .reduceToVectors(
    {geometry: URBAN_BORDER}
  );

//// LST URBAN

// LST MODIS urban night
var LST_URBAN_NIGHT = LST_NIGHT
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

```

```

// LST MODIS urban day
var LST_URBAN_DAY = LST_DAY
  .clip(URBAN_BORDER)
  .clip(CLC_URBAN);

//// LST RURAL

// Maximum elevation in urban areas
var MAX_VAL_URBAN_ELEVATION = ee.Number(
  SRTM
  .reduceRegion({
    reducer: ee.Reducer.max(), // max value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  }).get("elevation")
);

// Minimum elevation in urban areas
var MIN_VAL_URBAN_ELEVATION = ee.Number(
  SRTM
  .reduceRegion({
    reducer: ee.Reducer.min(), // min value
    geometry: CLC_URBAN, // polygon region
    scale: 1000 // 1000m
  }).get("elevation")
);

/// CLC RURAL

// CLC rural classes
var CLC_RURAL_RASTER = CLC
  .updateMask(CLC.gte(211).and(CLC.lte(335)));

// CLC raster to vector conversion
var CLC_RURAL = CLC_RURAL_RASTER
  .reduceToVectors(
    {geometry: RURAL_BORDER}
  );

/// SRTM RURAL

// SRTM elevation max difference <> +/- 50 meters urban areas
var SRTM_RURAL_AREAS_RASTER = SRTM

.updateMask(SRTM.lte(MAX_VAL_URBAN_ELEVATION.add(30)).and(SRTM.gte(MIN_VAL_URBAN_E
LEVATION.subtract(30))));

// CLC raster to vector conversion
var SRTM_RURAL = SRTM_RURAL_AREAS_RASTER
  .reduceToVectors(
    {geometry: RURAL_BORDER, scale: 1000}
  );

// LST MODIS rural night
var LST_RURAL_NIGHT = LST_NIGHT
  .clip(RURAL_BORDER)
  .clip(CLC_RURAL)
  .clip(SRTM_RURAL);

// LST MODIS rural day
var LST_RURAL_DAY = LST_DAY
  .clip(RURAL_BORDER)
  .clip(CLC_RURAL)
  .clip(SRTM_RURAL);

/// NDVI URBAN

var NDVI_URBAN = NDVI_MONTHLY_MEAN
  .clip(URBAN_BORDER)

```

```

        .clip(CLC_URBAN)
        .select("NDVI");

    /// SRTM URBAN

    var SRTM_URBAN = SRTM
        .clip(URBAN_BORDER)
        .clip(CLC_URBAN)
        .select("elevation");

    /// POPULATION URBAN

    var POPULATION_URBAN = POPULTAION_MONTHLY_MEAN
        .clip(URBAN_BORDER)
        .clip(CLC_URBAN)
        .select("population");

    /// NDWI URBAN

    var NDWI_URBAN = NDWI_MONTHLY_MEAN
        .clip(URBAN_BORDER)
        .clip(CLC_URBAN)
        .select("NDWI");

    /// LIGHT URBAN

    var LIGHT_URBAN = LIGHT_MONTHLY_MEAN
        .clip(URBAN_BORDER)
        .clip(CLC_URBAN)
        .select("avg_rad");

    /// ALBEDO URBAN

    var ALBEDO_URBAN = ALBEDO_MONTHLY_MEAN
        .clip(URBAN_BORDER)
        .clip(CLC_URBAN)
        .select("Albedo_WSA_shortwave");

    /// IMPERVIOUS SURFACE URBAN

    var IMPERVIOUS_SURFACE_URBAN = IMPERVIOUS_SURFACE
        .clip(URBAN_BORDER)
        .clip(CLC_URBAN)
        .rename("impervious_surfaces");

    /// LST RURAL MEAN VALUE

    // LST rural mean value night
    var LST_RURAL_MEAN_NIGHT = ee.Number(
        LST_K_to_C(LST_RURAL_NIGHT)
        .reduceRegion({
            reducer: ee.Reducer.mean(), // mean value
            geometry: CLC_RURAL.geometry(), // polygon region
            scale: 1000 // 1000m
        }).get("LST_Night_1km")
    );

    // LST rural mean value day
    var LST_RURAL_MEAN_DAY = ee.Number(
        LST_K_to_C(LST_RURAL_DAY)
        .reduceRegion({
            reducer: ee.Reducer.mean(), // mean value
            geometry: CLC_RURAL, // polygon region
            scale: 1000 // 1000m
        }).get("LST_Day_1km")
    );

    /// SUHI CALCULATION ///

```

```

// SUHI night
var SUHI_NIGHT = LST_K_to_C(LST_URBAN_NIGHT)
  .subtract(ee.Image.constant(LST_RURAL_MEAN_NIGHT))
  .rename("SUHI_NIGHT")
  .toFloat();

// SUHI day
var SUHI_DAY = LST_K_to_C(LST_URBAN_DAY)
  .subtract(ee.Image.constant(LST_RURAL_MEAN_DAY))
  .rename("SUHI_DAY")
  .toFloat();

////////////////////////////////////

if (i === 0) {

  print(CITIES[cities_number]);

}

print("MONTH: " + month);

////////////////////////////////////

/// WRITING TO ARRAYS

SUHI_MONTHS_NIGHT.push(SUHI_NIGHT);
SUHI_MONTHS_DAY.push(SUHI_DAY);

NDVI_MONTHS.push(NDVI_SCALED(NDVI_URBAN));
POPULATION_MONTHS.push(POPULATION_URBAN);
NDWI_MONTHS.push(NDWI_URBAN);
LIGHT_MONTHS.push(LIGHT_URBAN);
ALBEDO_MONTHS.push(ALBEDO_SCALED(ALBEDO_URBAN));

//// EXPORTING SAMPLES

if (i == MONTHS_TO_INCLUDE.length - 1) {

  IMPERVIOUS_SURFACES_CITIES.push(IMPERVIOUS_SURFACE_URBAN);
  URBAN_AREAS_GEOMETRIES.push(CLC_URBAN);

  if (cities_number == CITIES.length - 1) {

    var SUHI_DAY_AVERAGE_CITIES = ee.ImageCollection(SUHI_MONTHS_DAY).mean();
    var SUHI_NIGHT_AVERAGE_CITIES =
ee.ImageCollection(SUHI_MONTHS_NIGHT).mean();

    var NDVI_AVERAGE_CITIES = ee.ImageCollection(NDVI_MONTHS).mean();
    var POPULATION_AVERAGE_CITIES =
ee.ImageCollection(POPULATION_MONTHS).mean();
    var NDWI_AVERAGE_CITIES = ee.ImageCollection(NDWI_MONTHS).mean();
    var LIGHT_AVERAGE_CITIES = ee.ImageCollection(LIGHT_MONTHS).mean();
    var ALBEDO_AVERAGE_CITIES = ee.ImageCollection(ALBEDO_MONTHS).mean();
    var IMPERVIOUS_SURFACES_AVERAGE_CITIES =
ee.ImageCollection(IMPERVIOUS_SURFACES_CITIES).mosaic();

    var URBAN_AREAS = join_urban_areas(URBAN_AREAS_GEOMETRIES);

// JOINING BANDS
var BANDS_FACTORS_SUHI =
  SUHI_DAY_AVERAGE_CITIES
  .addBands(SUHI_NIGHT_AVERAGE_CITIES)
  .addBands(NDVI_AVERAGE_CITIES)
  .addBands(POPULATION_AVERAGE_CITIES)
  .addBands(NDWI_AVERAGE_CITIES)
  .addBands(LIGHT_AVERAGE_CITIES)

```

```

        .addBands(ALBEDO_AVERAGE_CITIES)
        .addBands(IMPERVIOUS_SURFACES_AVERAGE_CITIES);

    print(BANDS_FACTORS_SUHI);

    // SAMPLES
    var SAMPLES_FACTORS_SUHI = point_sampling(BANDS_FACTORS_SUHI, URBAN_AREAS,
1000);

    // EXPORTING
    Export.table.toDrive({
        collection: SAMPLES_FACTORS_SUHI,
        description: 'image_sample_demo_SUHI',
        folder: 'earth_engine_demos',
        fileFormat: 'CSV'
    });

    Map.addLayer(SAMPLES_FACTORS_SUHI);
}
}
}
}

//// VIEW OF EUROPE
Map.setCenter(14.869854282951186, 48.69447525861791, 5);

```

Příloha č. 4: Absolutní standardizované regresní koeficienty

Model		absolutní standardizované regresní koeficienty						
		Albedo (WSA)	NDVI	NDWI	Záření světél	Nadmořská výška	Nepropustnost povrchu	Hustota obyvatel
Amsterdam	SUHII denní	0	0	0	0.406	0	0	0.338
	SUHII noční	0.574	0.277	0	0	0	0	0.528
Berlín	SUHII denní	0.33	0.373	0	0	0.09	0	0.43
	SUHII noční	0.335	0.413	0	0.266	0	0	0.257
Budapešť	SUHII denní	0.302	0	0.266	0	0.274	0.389	0.159
	SUHII noční	0.304	0	0.28	0.11	0	0.271	0.258
Londýn	SUHII denní	0	0.409	0.393	0	0.334	0.271	0.253
	SUHII noční	0.491	0	0.092	0	0.18	0	0.15
Madrid	SUHII denní	0.315	0.247	0.097	0	0.214	0	0.357
	SUHII noční	0.351	0.526	0.196	0.127	0	0	0.434
Paříž	SUHII denní	0.083	0.508	0.146	0.114	0.268	0.161	0.286
	SUHII noční	0.415	0	0	0.12	0.072	0.089	0.347
Praha	SUHII denní	0.34	0.322	0.164	0.163	0.146	0	0.446
	SUHII noční	0.496	0.128	0	0.145	0.155	0	0.277
Řím	SUHII denní	0.517	0.948	0.384	0.19	0	0	0.208
	SUHII noční	0.37	0.346	0.169	0.226	0	0.098	0.138
Stockholm	SUHII denní	0.325	0	0.223	0.524	0.12	0	0
	SUHII noční	0.483	0	0.147	0.2	0.328	0	0.171
Varšava	SUHII denní	0.205	0.344	0	0	0.202	0.18	0.213
	SUHII noční	0.317	0.434	0.29	0	0.15	0	0.296