

IT-kwaliteitscriteria

Acquisitie van de data

Om de data van de verschillende afdelingen te verzamelen, zullen we drie aparte R-scripts gebruiken. Voor afdeling A, waar data worden geleverd in een Excel-bestand, kunnen we een R-script schrijven om de data automatisch uit te lezen en relevante informatie te extraheren. Voor afdeling B, die haar statistieken publiceert op Statline, zullen we een ander R-script gebruiken om de gegevens rechtstreeks via de beschikbare API op te halen. En voor afdeling C, waar we toegang hebben tot een SQL-database, zullen we een derde R-script gebruiken om verbinding te maken met de database en bijvoorbeeld m.b.v. een SQL query de data op te halen.

Integreren van de ontvangen data

Nadat we de data van alle afdelingen hebben verzameld, kunnen we ze combineren in één dataframe m. b.v. een R-script. Voordat we de gegevens combineren, moeten we ervoor zorgen dat ze genormaliseerd zijn en eventuele inconsistenties worden aangepakt zodat we alle drie onder een soortgelijk kop kunnen brengen als het nodig is. Vervolgens zullen we de benodigde berekeningen uitvoeren om het gewenste eindcijfer te verkrijgen. Deze “streamlined” aanpak minimaliseert handmatige invoer en vermindert de kans op fouten, terwijl de reproduceerbaarheid van het proces wordt vergroot.

Reproduceerbaarheid

Om ervoor te zorgen dat het proces op een consistente en betrouwbare manier kan worden herhaald, ongeacht waar en wanneer het wordt uitgevoerd, zullen we gebruikmaken van projectstructuren in RStudio om alle scripts, data en documentatie georganiseerd te houden binnen één projectmap. Door de vereiste pakketten en hun versies zorgvuldig te beheren, kunnen we ervoor zorgen dat het project op verschillende machines reproduceerbaar blijft, wat de consistentie en betrouwbaarheid van de resultaten bevordert.

Documentatie

Het is essentieel om elk script en elke stap van het proces uitgebreid te documenteren met duidelijke opmerkingen en beschrijvingen. Daarnaast moeten we ook de gebruikte bronnen, waaronder API-documentatie, datamodellen en SQL-query's, documenteren om een volledig inzicht te geven in de herkomst en verwerking van de gegevens.

Versiebeheer

Voor versiebeheer zullen we een versiebeheersysteem zoals Git gebruiken om wijzigingen in de code en documentatie bij te houden. Door regelmatig commits te maken en branches te gebruiken, kunnen we aan nieuwe functies werken zonder de hoofdcode te verstoren, waardoor de ontwikkeling van het project georganiseerd blijft.

Foutgevoeligheid

Om de foutgevoeligheid te minimaliseren, zullen we foutafhandeling toevoegen aan de scripts om onverwachte situaties, zoals ontbrekende gegevens of fouten bij het lezen van bestanden, te verwerken. Dit kan worden bereikt door try-catch blokken te gebruiken en specifieke foutmeldingen te genereren. Daarnaast zullen we controlemechanismen implementeren, zoals validatiechecks, om ervoor te zorgen dat de ingevoerde gegevens voldoen aan verwachte patronen en waarden, waardoor de betrouwbaarheid en nauwkeurigheid van de resultaten worden verhoogd.