



Testing in Python - Aufgaben

1. Vorbereitung

Bevor wir mit den Aufgaben starten können, müssen wir zunächst die Infrastruktur aufbauen:

1. Dein Kollege hat dieses Projekt mit der Python-Version 3.7.3 gelöst. Es wäre also gut, wenn du diese Version auch hast.
2. Lade dir dann das git-Verzeichnis [git@git.cc-demo.com:tutorials/python/Python_Tutorial.git](https://git.cc-demo.com:tutorials/python/Python_Tutorial.git) herunter. Gehe hierfür zu VCS  Get from Version Control und gebe obigen URL ein.
3. Öffne den Ordner Teil_4_Testing/aufgaben_testing/.
4. Installiere Pytest (pip install pytest) und Coverage (pip install pytest coverage). Weitere Informationen unter <https://docs.pytest.org/en/latest/getting-started.html>.
5. Es kann sein, dass du außerdem das Package Pandas importieren musst. Gehe in PyCharm Prefereces → Project: Python_Tutorial  Project Interpreter und klicke auf das +, nun kannst du Pandas hinzufügen.
6. Tests lassen sich über, wie im Tutorial erwähnt, beispielsweise über den folgenden Befehl ausführen (Navigiere zuerst in den Ordner Teil_4_Testing/aufgaben_testing/test/Unit):

```
python3 -m pytest -v --cov-report html:my_tests_results/py_cov.html -  
cov=project_dir\Python_Tutorial\Teil_4_Testing\aufgaben_testing\scripts -  
html=my_tests_results/py-test-results.html tests.py
```
7. Jetzt kannst du mit der Bearbeitung starten!

2. Beschreibung

Es soll ein Skript entwickelt werden, das Daten einliest, formatiert und zusammenfügt. Auf diesen Datensatz sollen Durchschnitte auf Gruppenausprägungen berechnet werden. Diese müssen anschließend ausgeschrieben werden. Dazu stehen folgende Datensätze zur Verfügung:

- Aberkauf.xlsx - wöchentliche Abverkäufe pro Material und Sparte (z.B. Filialen).
- Stammdaten.csv – verschiedene Merkmale pro Material.

Gott sei Dank hat ein Mitarbeiter schon Vorarbeit geleistet und einige Funktionen geschrieben und mit Tests versehen. Dir stehen deshalb zur Bearbeitung schon drei Skripte zur Verfügung:

- mainScript.py - Hier soll der Code am Ende ausgeführt werden.
- functions.py - Hier findest du alle Funktionsdefinitionen.
- tests.py - Hier findest du alle Tests zu den in functions.py stehenden Definitionen.

Bitte achte bei der Bearbeitung auch auf unsere Codingstandards und insbesondere auf Folgendes:

- Jede Funktion benötigt eine vollständige Beschreibung.
- Jede Funktion benötigt die passenden Tests.
- Jede Funktion muss verwendet werden.

- Skript functions.py sollte alphabetisch geordnet sein.
- Skript tests.py sollte alphabetisch geordnet sein.

Ordnerstruktur:

```
aufgaben_testing
+---scripts
    +-- mainScript.py
    +-- mainScript_musterloesung.py
    +-- functions.py
    +-- functions_musterloesung.py
+-- test
    +-- Unit
        +-- tests.py
        +-- tests_musterloesung.py
        +-- results
        +-- test_data
        +-- data
```

3. Aufgabe

Dir wurden nun folgende Aufgaben aufgetragen:

1. Schaue dir den bisher ausgearbeiteten Code genau an.
2. Da der Code rechenintensiv ist, soll er direkt abbrechen, wenn entweder keine Abverkaufs- oder keine Stammdaten zur Verfügung stehen. Erweitere deshalb sowohl die Funktion readSales() als auch die Funktion readStamm() mit den zusätzlichen Prüfungen sowie den zugehörigen Tests.
3. Die formatierten Datensätze sales und stamm sollen nun verbunden werden. Definiere dafür eine Funktion mergeStammToSales() mit den Inputparametern sales und stamm. Die Daten sollen anhand der Dimension Material verbunden werden. Falls in den sales Daten eine Ausprägung von Material auftaucht, die keine Stammdaten enthält, muss abgebrochen werden mit einer aussagekräftigen Fehlermeldung. Erstelle zu der Funktion auch die passenden Tests. Füge dann die Funktion in das Skript ein und übergebe deren Output an die Variable mergedData.
4. Versuche dich nun in TDD (Test Driven Development). Die folgende Funktion soll so allgemein wie möglich geschrieben werden, da wir diese vielleicht in anderen Projekten noch benötigen. Übergabewerte sind ein Dataframe (pandas.DataFrame), ein String meanColumn und ein String aggregateColumns. Es sollen dann die Durchschnitte pro den in aggregateColumns Gruppen auf die Spalte meanColumn berechnet werden. Überlege dir, welche Tests hierfür relevant sind, schreibe die Tests und definiere dann die Funktion. Im Aufruf soll dann für das Dataframe mergedData pro Material und Lieferant die Durchschnitte für sale ausgerechnet werden.