# Aufgabe 1 & 2 in Python

**Aufgabe 1 – Aufgabenstellung:**

**a.** Werten Sie alle in den Daten vorkommenden Features (Feature 1 bis Feature 6) deskriptiv aus. Bitte beachten Sie, dass sich mit den Städtenamen (Tabelle Adresse) und den Datumsangaben 6 verschiedene Gruppen bilden. Berechnen Sie den Mittelwert, den Median, die Standardabweichung, den minimalen und maximalen Wert sowie die 25% und 75% Quantile. Erstellen Sie für jedes der Features je eine Grafik mit 6 Histogrammen für die Gruppen (horizontal die 3 Städte, vertikal das Datum). Hinweis: verwenden Sie die subplots Funktion aus der matplotlib Library, Beispiel: <https://matplotlib.org/examples/pylab_examples/subplots_demo.html>

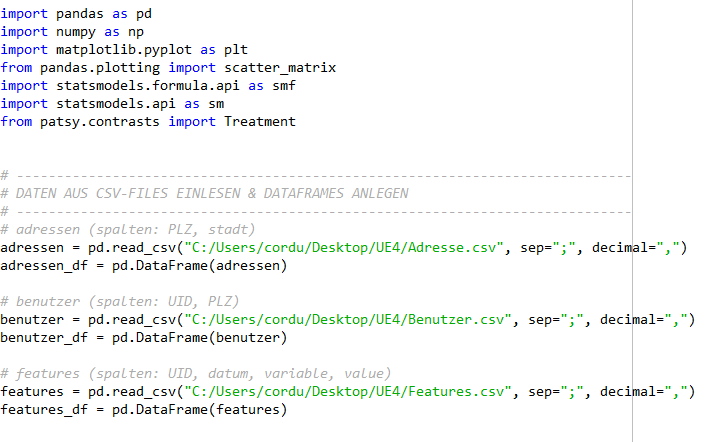
**b.** Gibt es fehlende Werte in dem Datensatz? Wenn ja, wie viele? Können Sie diese Daten imputieren? Falls Sie die Daten nicht imputieren können, dann entfernen Sie unvollständige Datensätze aus dem Analysebestand.

**c.** Visualisieren Sie die Korrelationsmatrizen (z.B. unter Verwendung der matshow Funktion aus der matplotlib Bibliothek).

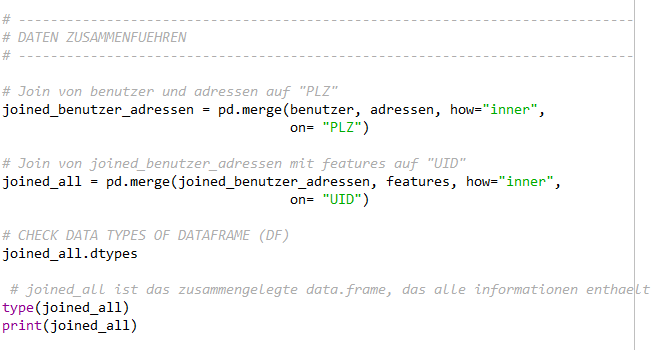
**d.** Dokumentieren Sie die notwendigen Schritte zur Datenaufbereitung.

In dieser Dokumentation wird der lauffähige Python-Code für die Lösung von Aufgabe 1 & 2 erklärt und an den jeweiligen Stellen werden die Analyseergebnisse besprochen. Für den gesamten Code zur Ausführung siehe Python-File, das ebenfalls auf Moodle hochgeladen wurde.

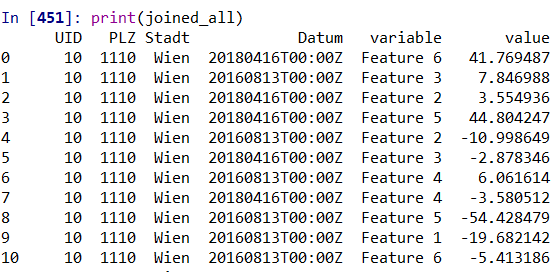
***Im Folgenden wir der Python-Code erklärt und dokumentiert:***



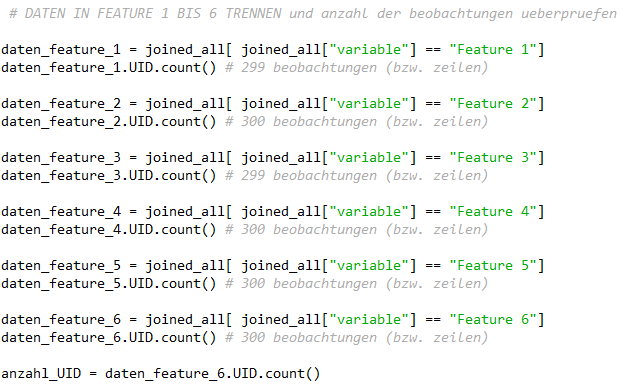
Zunächst werden die notwendigen Libraries (*pandas*, *numpy*, *matplotlib*, *statsmodels*) importiert. Dann werden die Daten aus den 3 csv-Files mittels der pd.read\_csv-Funktion von *pandas* eingelesen. Die eingelesenen Daten werden als Dataframe gespeichert.



Dann werden die Daten der 3 obigen Dataframes „benutzer\_df“, „features\_df“ und „adressen\_df“ zusammengeführt mit Hilfe von pd.merge aus pandas, sodass sich das Dataframe „joined\_all“ ergibt, das folgendermaßen aussieht:



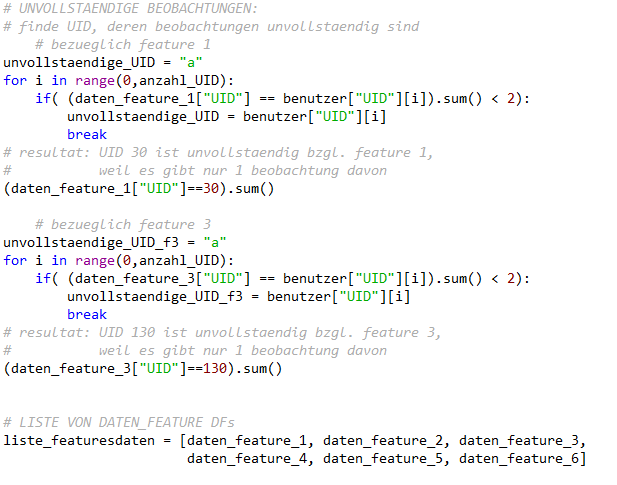




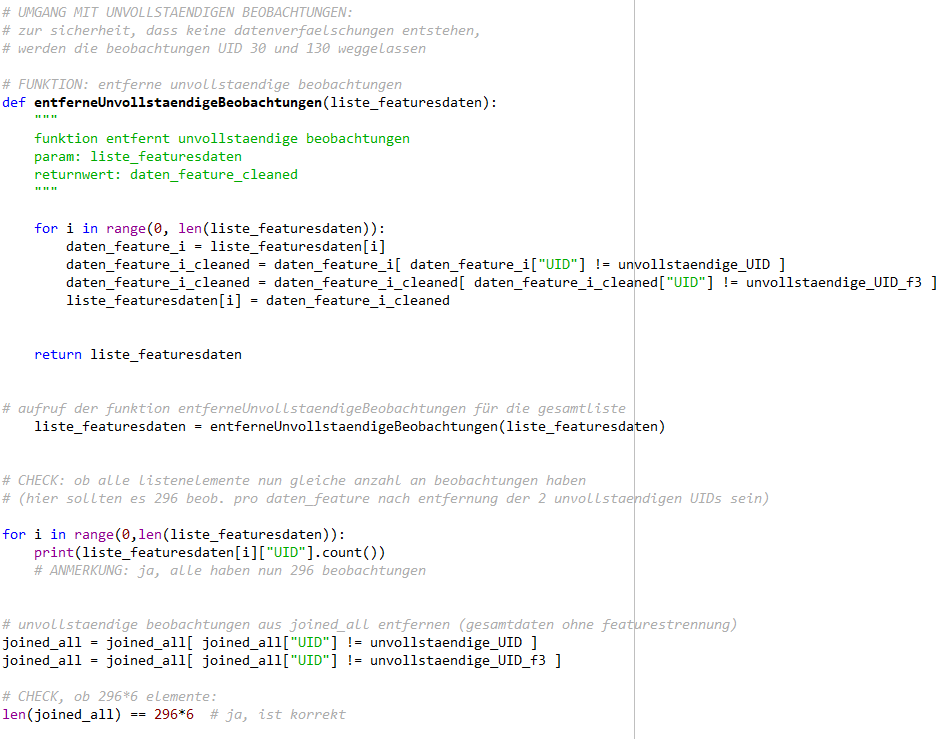
Nun werden die Daten anhand der Spalte „variable“ des Dataframe *joined*\_*all* in die einzelnen Features (1-6) getrennt.

Wie schon aus den Kommentaren zu der Anzahl der Beobachtungen erkennbar, wurde hier festgestellt, dass scheinbar für jeden Benutzer (UID) 2 Beobachtungen pro Feature vorhanden sein sollte. Allerdings ist dies nicht der Fall, denn Feature 1 und Feature 3 haben jeweils nur 1 Beobachtung bei bestimmten UIDs. Diese UIDs gilt es zu ermitteln.

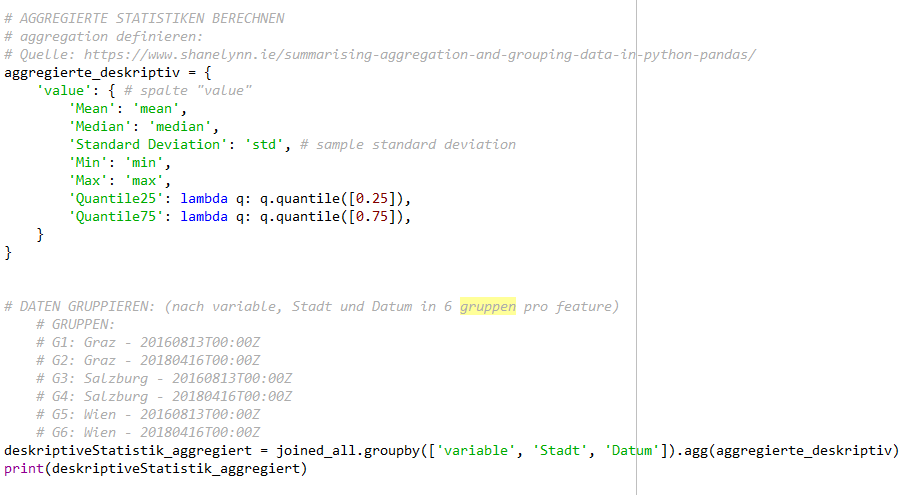
Der Umgang mit den fehlenden Werte kann entweder insofern erfolgen, dass man den „value“ beispielsweise durch den Median der Daten bzw. der Featuresdaten schätzt. Alle Daten außer dem „value“ können aus den restlichen Daten des betroffenen Benutzers übernommen werden. Andererseits kann man auch die betroffenen UIDs ganz weglassen aus allen Features, d.h. die Beobachtungen werden einfach aus dem Dataframe insgesamt herausgenommen.



Mittels der oben angeführten Code-Zeilen können die UIDs der unvollständigen Beobachtungen ermittelt werden. Es kann festgestellt werden, dass dies UID 30 bei Feature 1 ist, und UID 130 bei Feature 3.

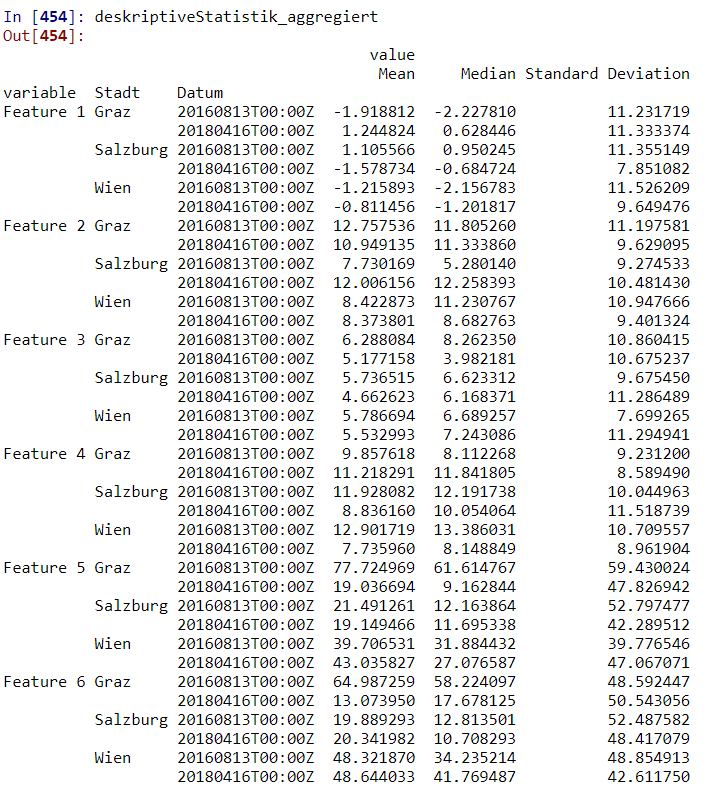


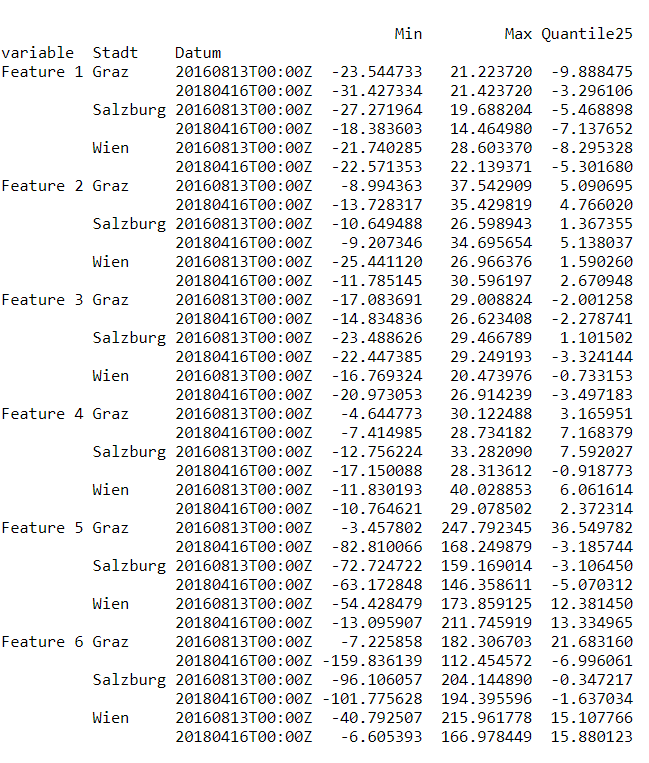
Die Funktion „*entferneUnvollstaendigeBeobachtungen*“ entfernt für eine übergebene Liste von Featuresdaten die unvollständigen Beobachtungen anhand der unvollständigen UID Informationen. Dann wird überprüft, ob tatsächlich nach Entfernen alle Features 296 Beobachtungen haben – dies wurde bestätigt.

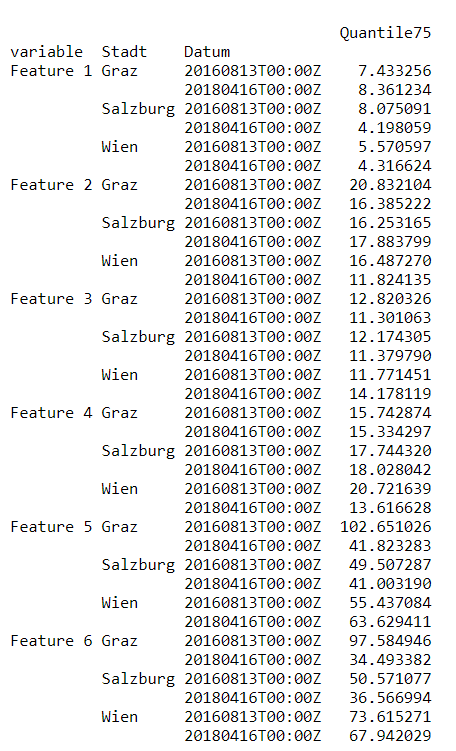


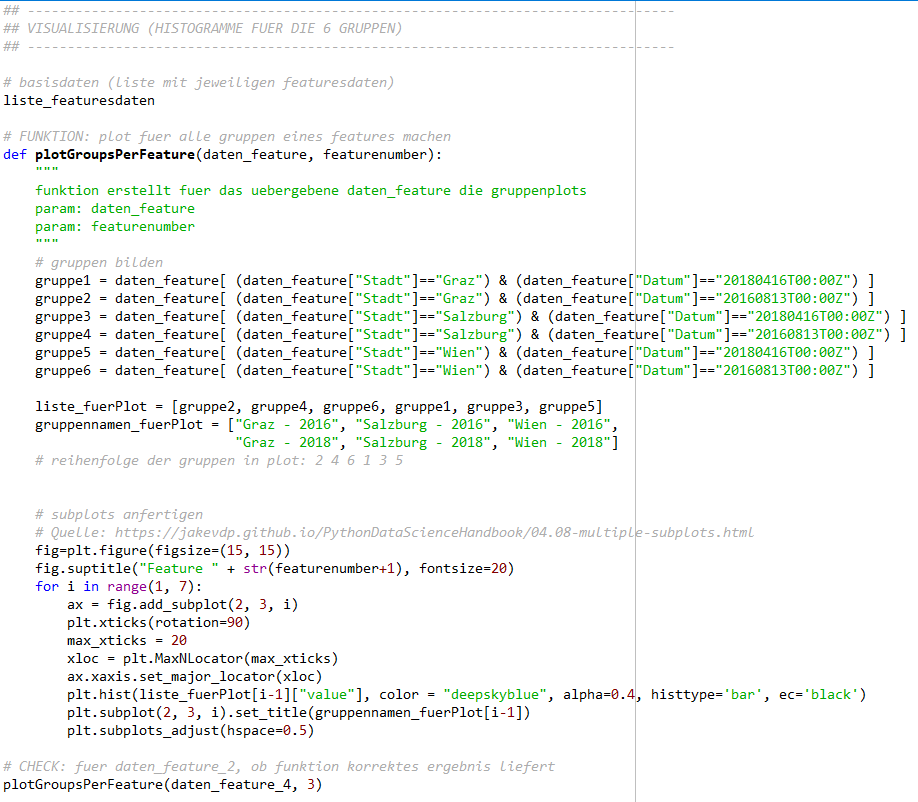
Für die Berechnung der deskriptiven Statistiken (Mittelwert, Median, Standardabweichung, Minimum, Maximum, 25% Quantil und 75% Quantil) pro Gruppe wird zuerst die Variable „*aggregierte\_deskriptiv*“ angelegt. Dann wird auf den mittels „*groupby*“ gruppierten Daten basierend auf „*joined\_all*“ die aggregierte Berechnung der deskriptiven Statistiken durchgeführt.

***Output der berechneten deskriptiven Statistiken pro Gruppe:***

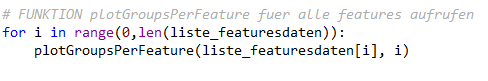




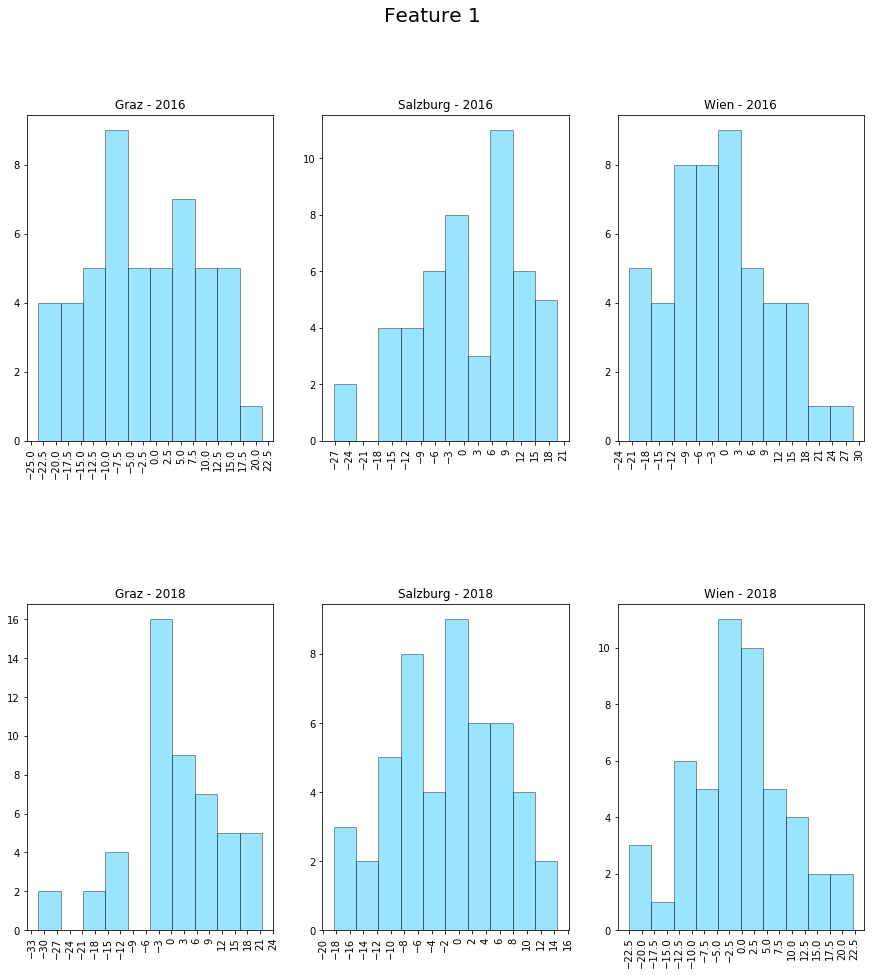


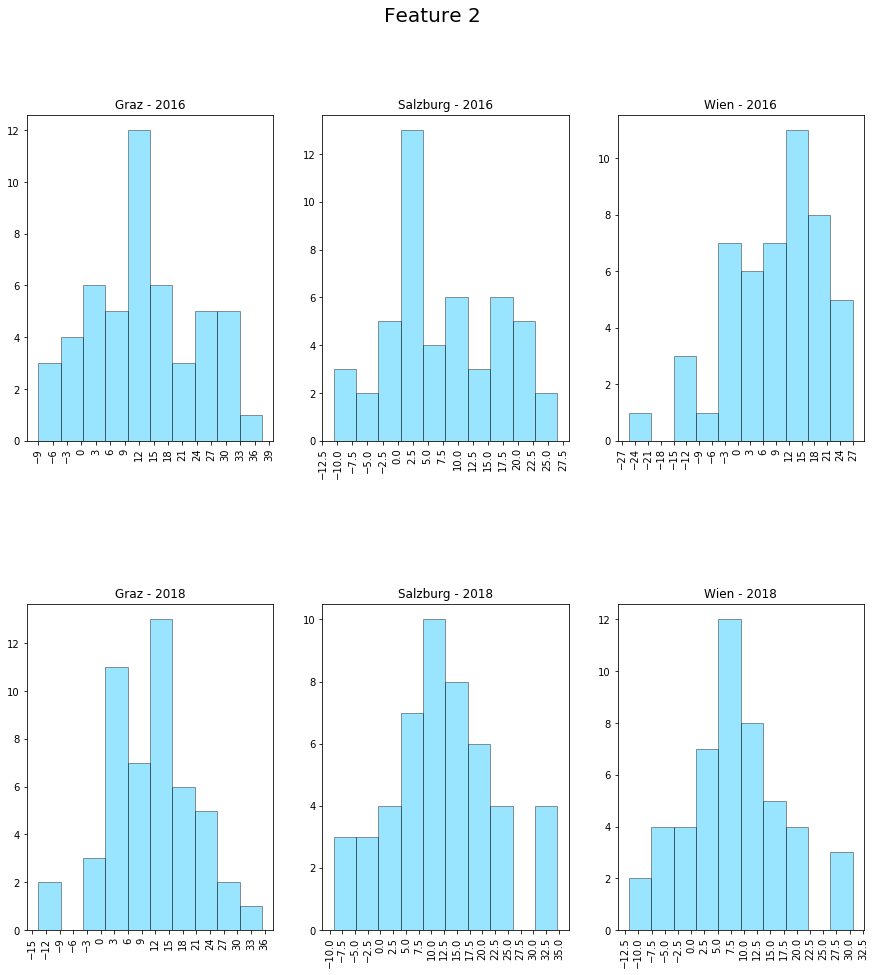


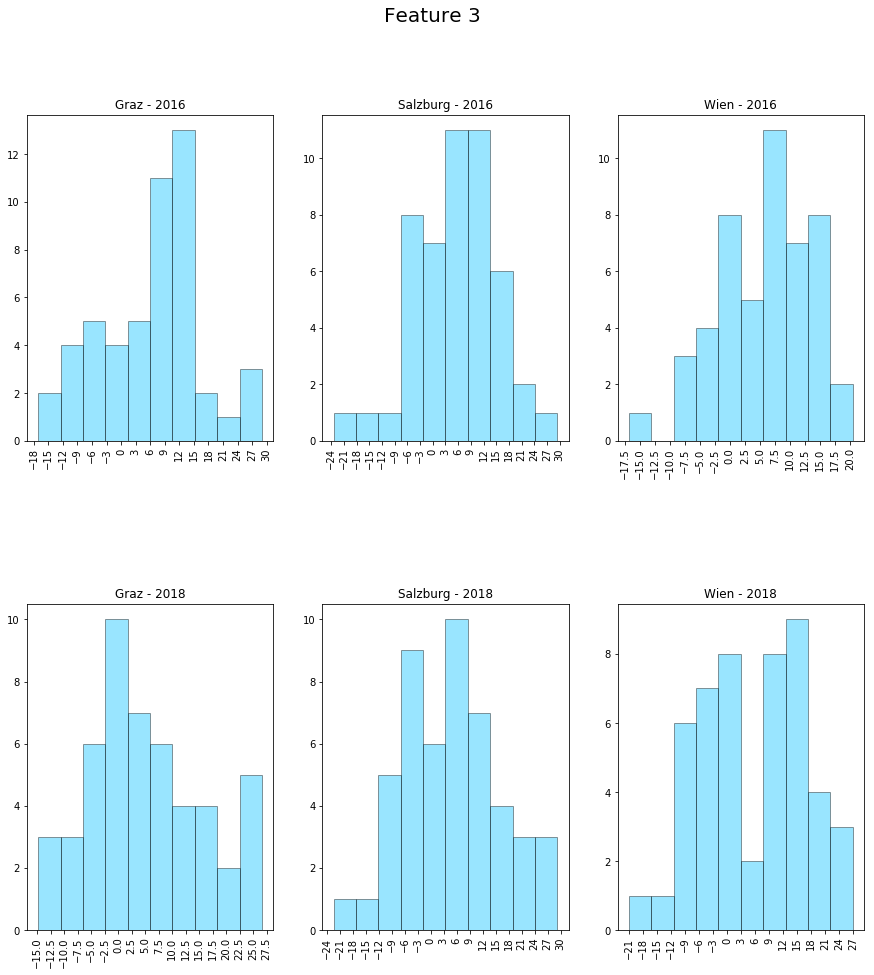
Zur Visualisierung der Histogramme pro Gruppe der jeweiligen Features dient die Funktion *plotGroupsPerFeature*, die „*daten\_feature*“ und „featurenumber“ als Parameter nimmt. Die Funktion bildet die 6 Gruppe innerhalb des Features und erstellt mittels „*plt.figure*“ und „*subplot*“ die 6 Plots, eingeteilt in Stadtinformationen horizontal und Datumsinformationen vertikal.

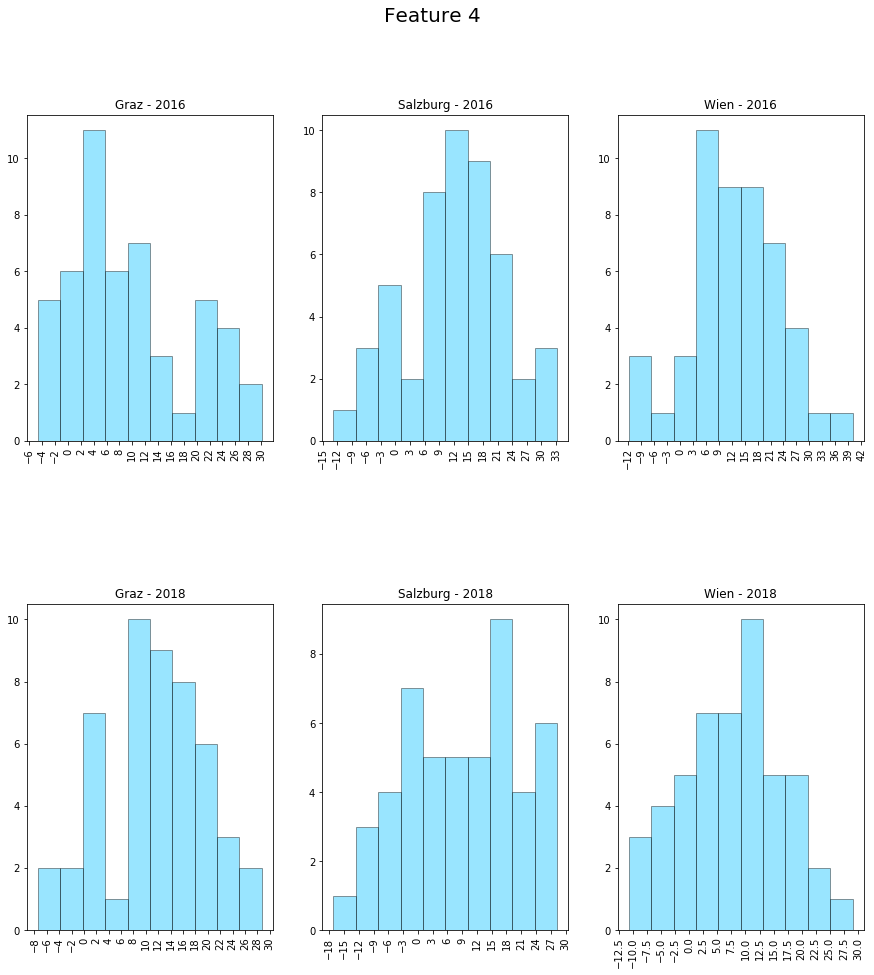


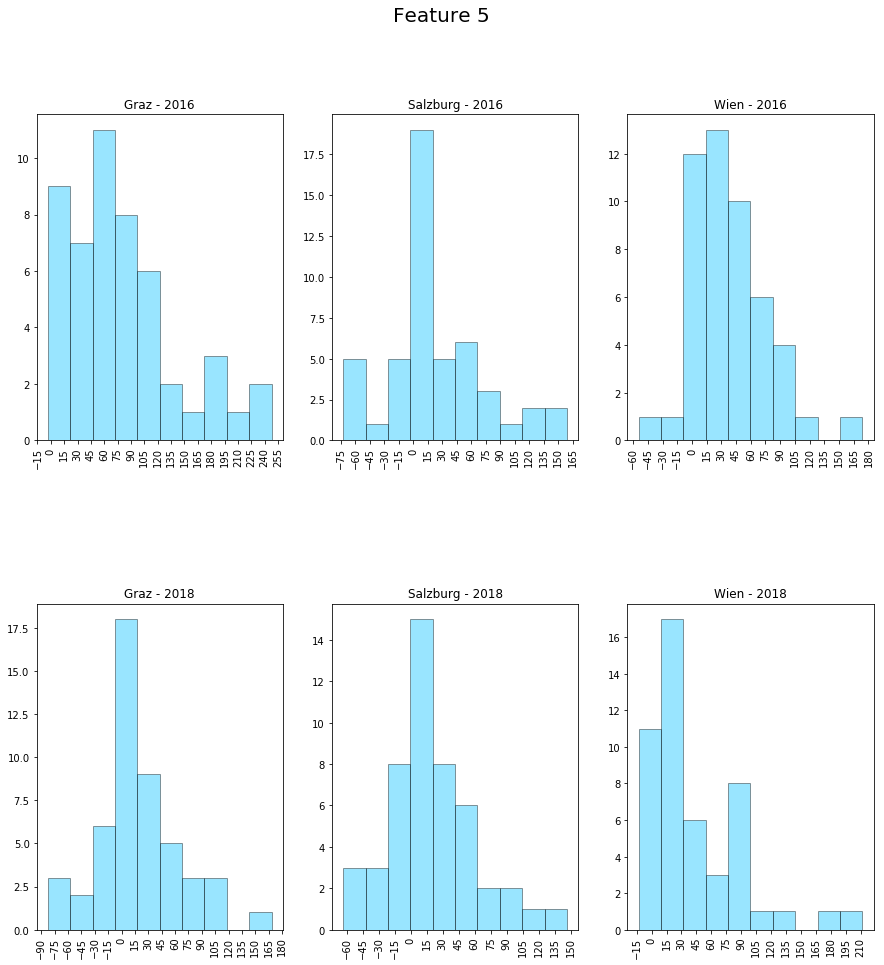
Die Funktion wird für die gesamte Liste an Featuresdaten aufgerufen, was folgende 6 Histogramm-Plots in der laut Angabe gewünschten Anordnung pro Feature generiert:

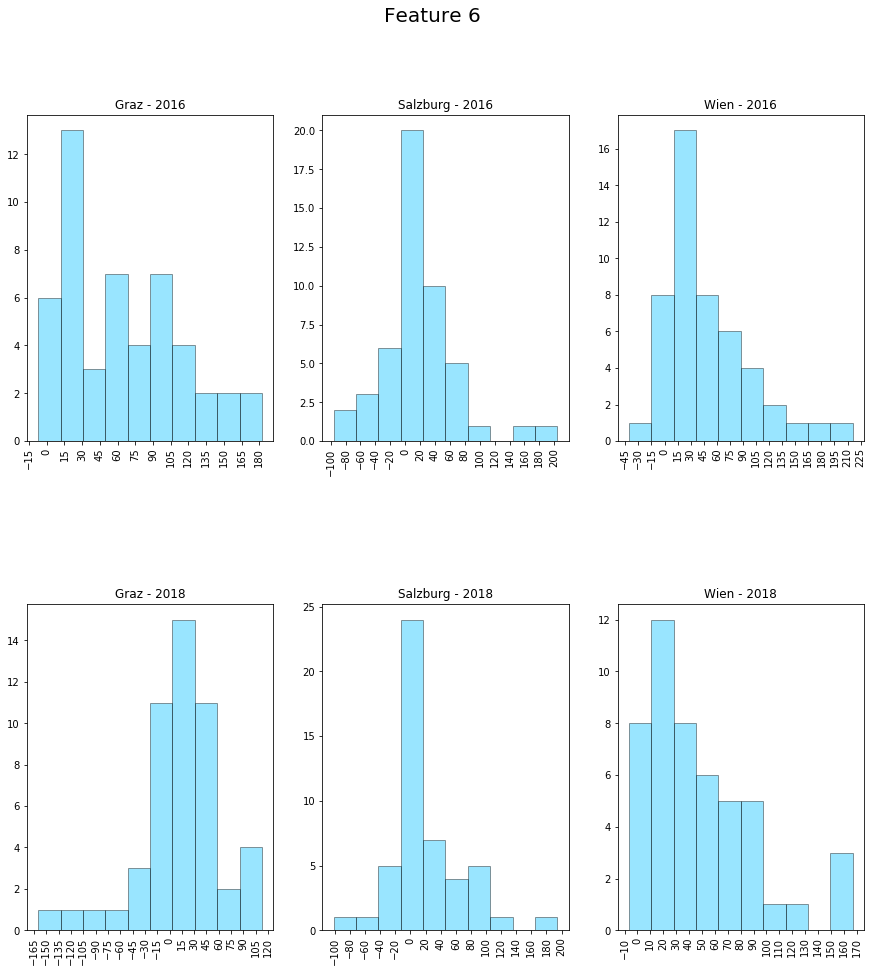


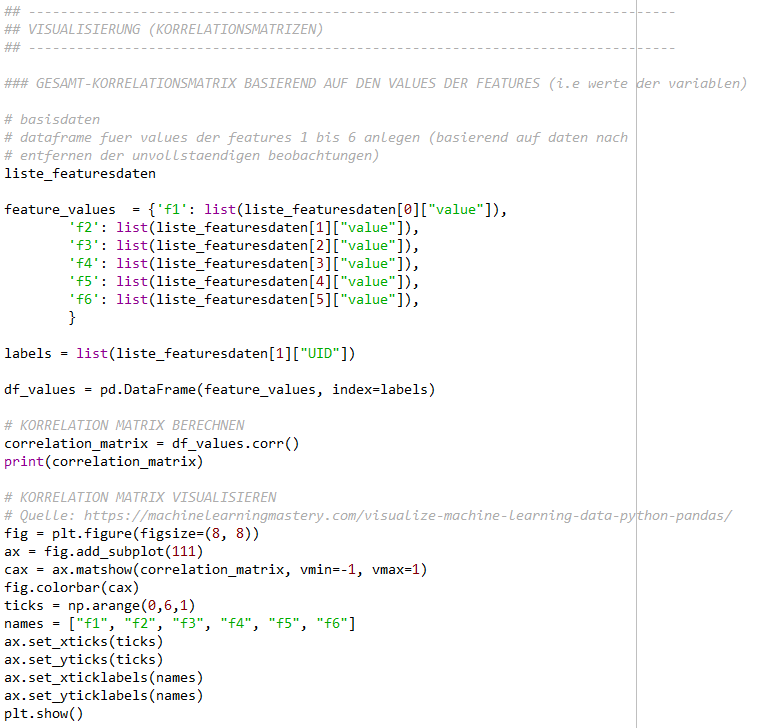






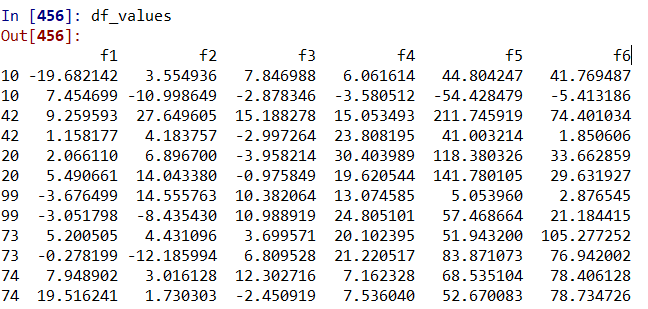




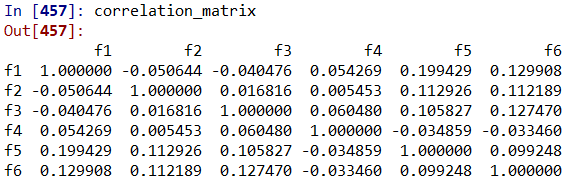


Für die Visualisierung der Korrelationsmatrizen wird ein Dataframe aus den „values“ der Features gebildet. Darauf wird die Funktion corr() aufgerufen, um die Korrelationsmatrix zu berechnen. Diese wird mittels „plt.figure“ und „ax.matshow“ geplottet.

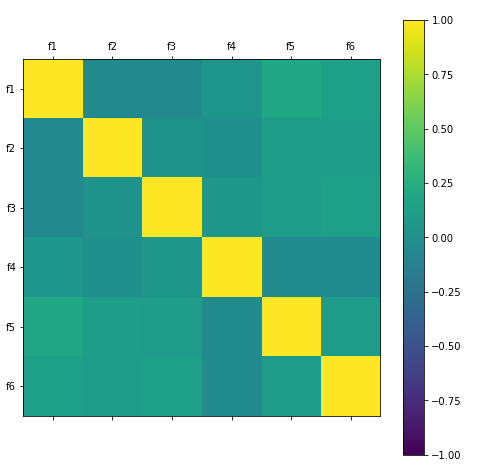
Das Dataframe, das die für die Berechnung notwendigen Feature-Values enthält, sieht folgendermaßen aus:



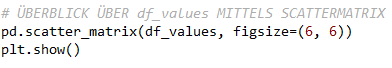
***Die Korrelationsmatrix:***

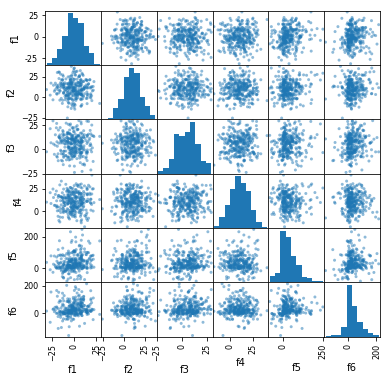


***Die Visualisierung der Korrelationsmatrix:***

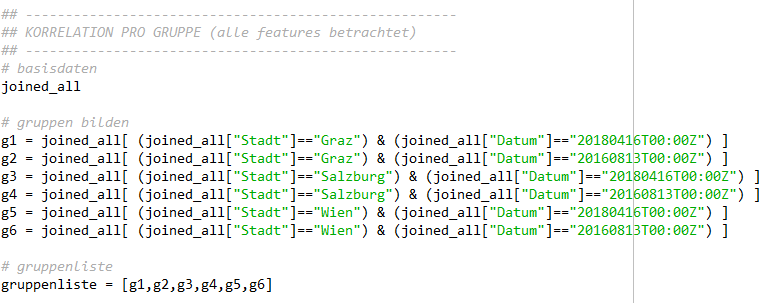


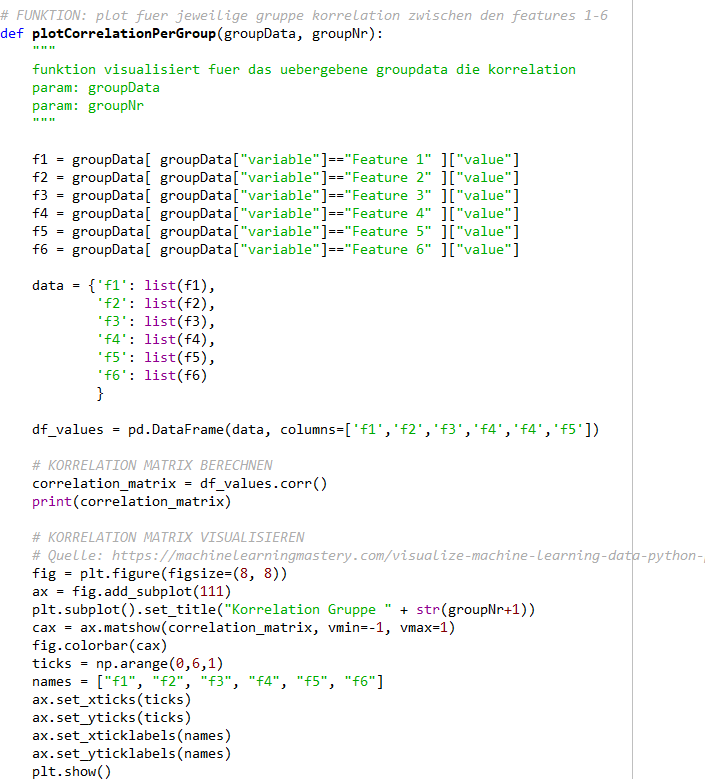
Die folgende Abbildung zeigt das Ergebnis der Darstellung von den Features f1 bis f6 mittels pandas *scatter\_matrix*.

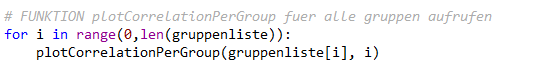




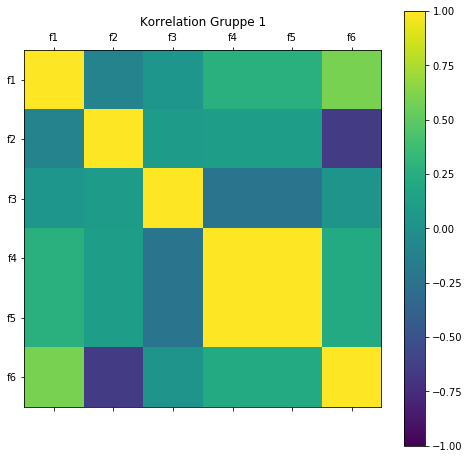
**Korrelation pro Gruppe (für jeweils alle Features):**

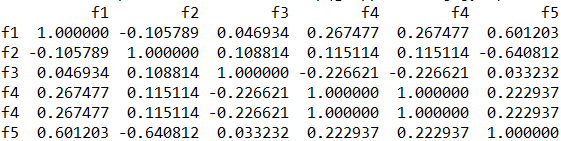


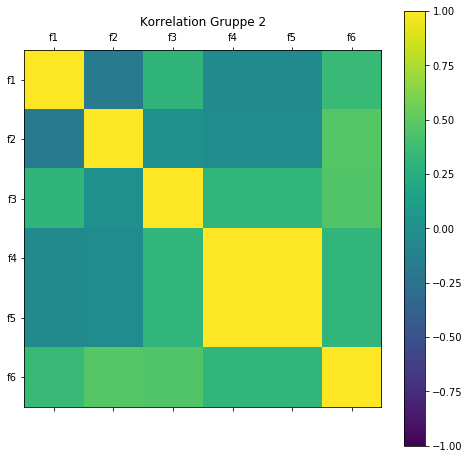


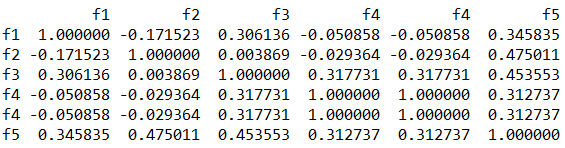


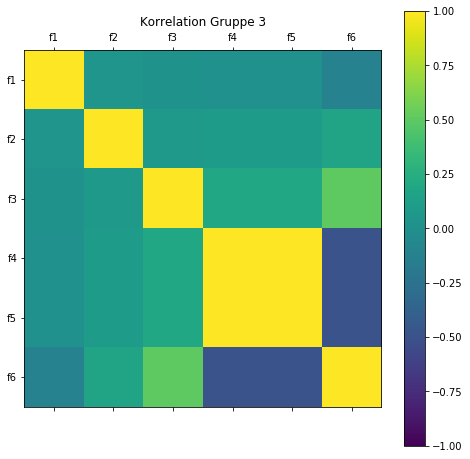
Visualisierung:

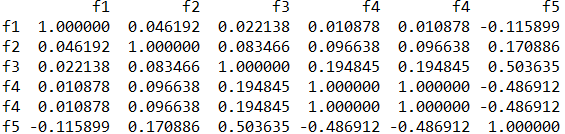


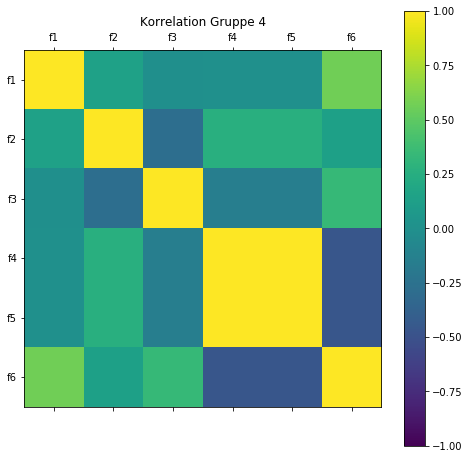


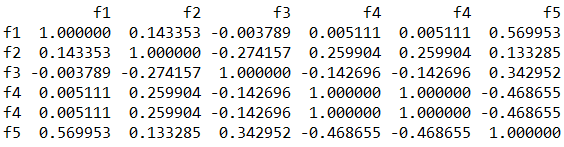


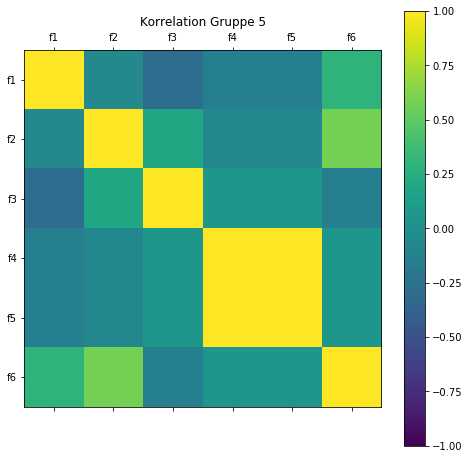


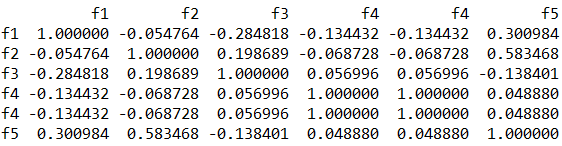


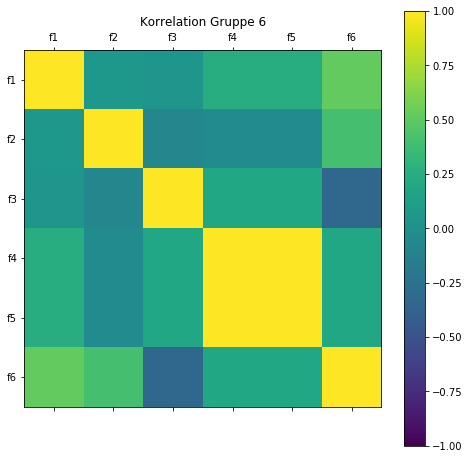


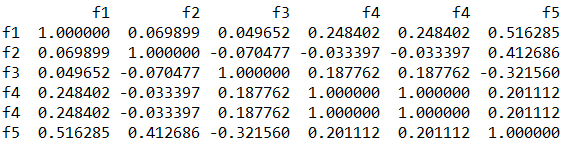












Anmerkung:

Die Korrelation zwischen den Features 1-6 ist je nach Gruppe unterschiedlich.

**Aufgabe 2 – Aufgabenstellung:**

a. Erzeugen Sie eine abgeleitete Variable aus der Summe von Feature 5 und Feature 6

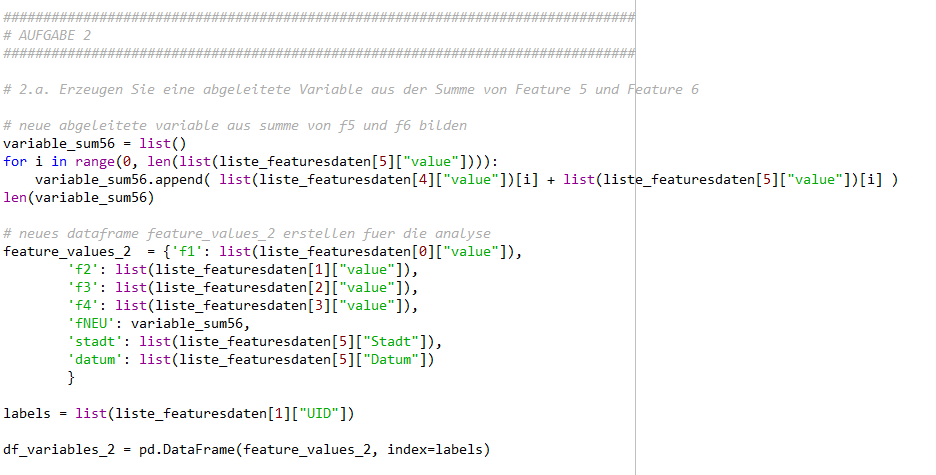
b. Gibt es Korrelationen zwischen den verbleibenden Variablen und der neuen abgeleiteten Variable?

c. Modellieren Sie die abgeleitete Variable mit einem linearen Modell.

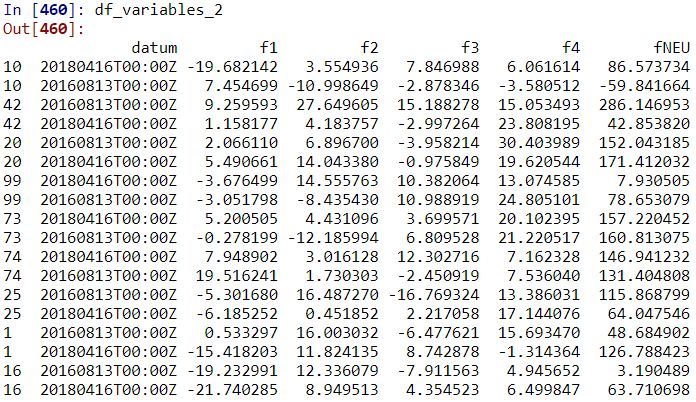
d. Welche Variablen sind im Modell sinnvoll, wie gehen Sie mit den kategoriellen Variablen um?

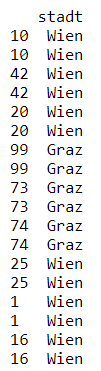
e. Beschreiben Sie Ihre Modellierungsergebnisse und erzeugen Sie Grafiken um Ihre Ergebnisse zu dokumentieren.

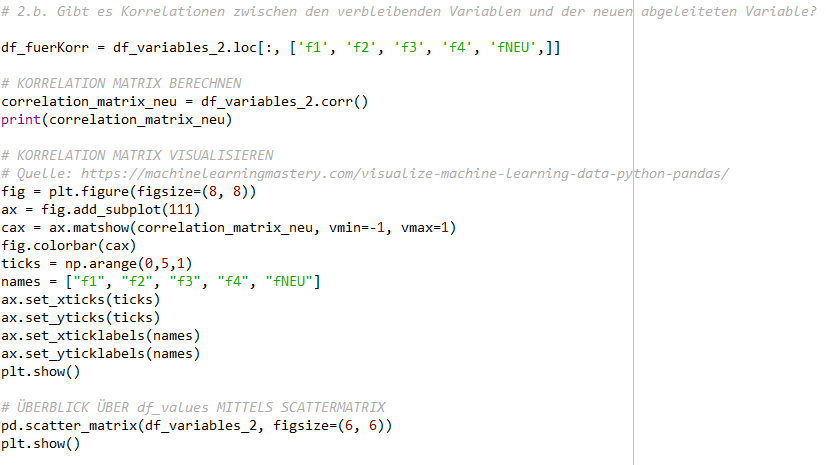
f. Welche Modellierungsmethode verwenden Sie und warum haben Sie sich für dieses Modell entschieden?



Aus der Summe von Feature 5 und Feature 6 wurde eine neue Variable namens „*variable\_sum56*“ erstellt. Dann wurde ein neues Dataframe, das die Spalten „*f1*“, „*f2*“, „*f3*“, „*f4*“, „*fNEU*“, „*stadt*“ und „*datum*“ enthält, erstellt. Die neue Variable wurde „*fNEU*“ genannt.

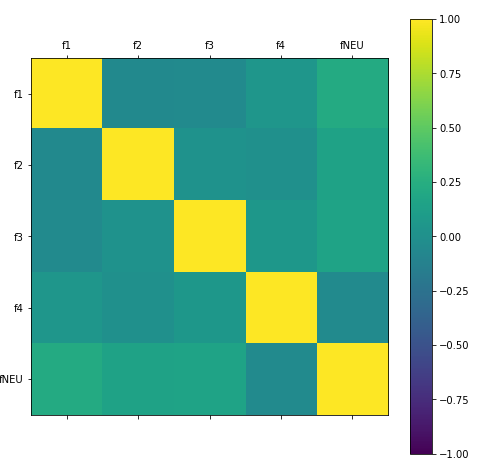
Das neue Dataframe sieht folgendermaßen aus:



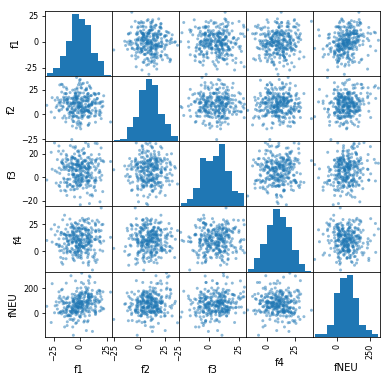


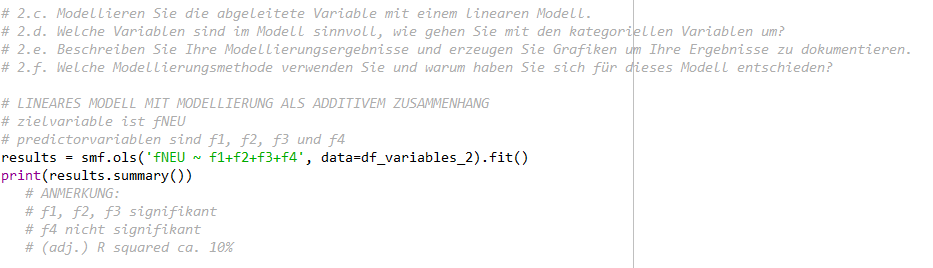
Die Korrelation zwischen den verbleibenden Variablen wird ebenfalls mit einer Korrelationsmatrix untersucht. Dafür wird die Funktion *corr()* verwendet, die die Korrelationsmatrix berechnet. Diese wird mittels „*matshow*“ geplottet.

***Die Korrelationsmatrix inkl. der neuen abgeleiteten Variable („fNEU“):***

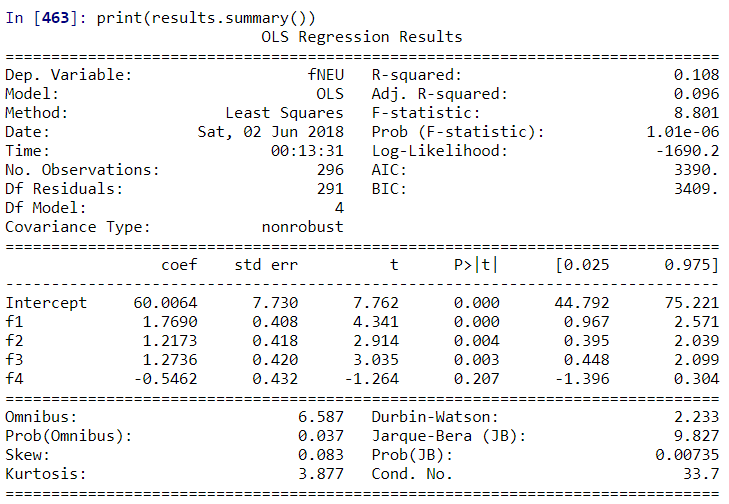


***Scatterplot inclusive der neuen abgeleiteten Variable („fNEU“):***

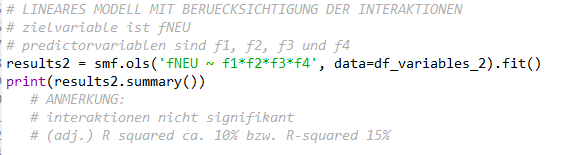




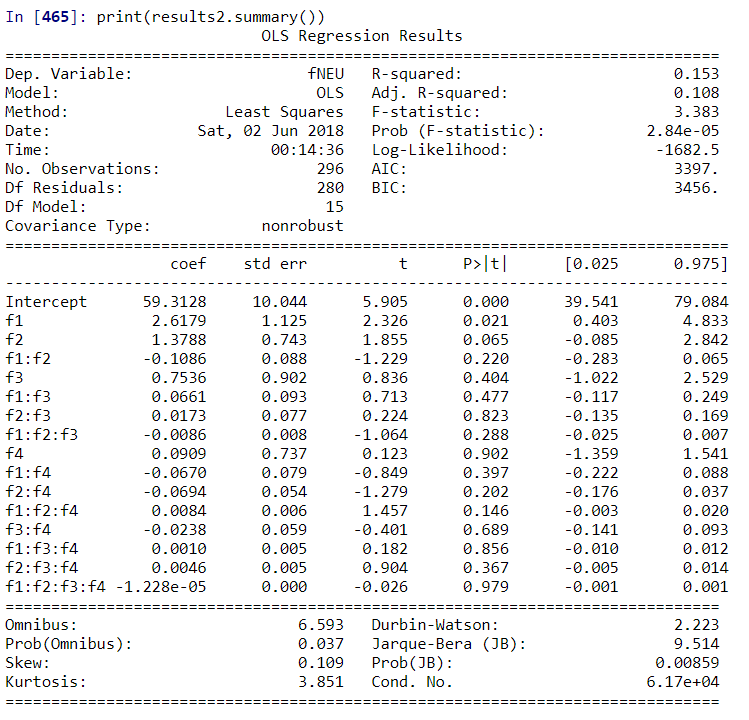
Als erster Versuch wird ein **lineares Modell (OLS) mit additivem Zusammenhang zwischen f1, f2, f3 und f4** zur Erklärung der Variable fNEU berechnet. Die erklärte Varianz (R-Squared bzw. adj. R-Squared) liegt bei 10% bzw. 11%. Die Variablen f1, f2 und f3 sind signifikant, der Variable f4 ist nicht signifikant beitragend zur Erklärung der Variable fNEU. Die Zusammenfassung der Resultate ergibt Folgendes:

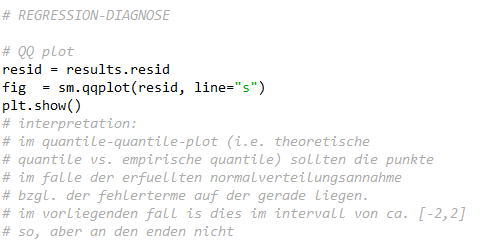


-----------------------------------------------------------------------------------------------------------------

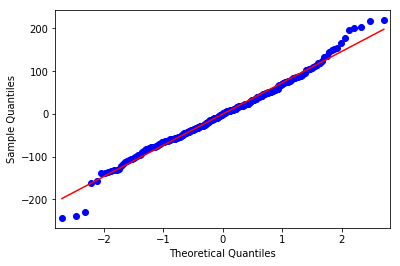


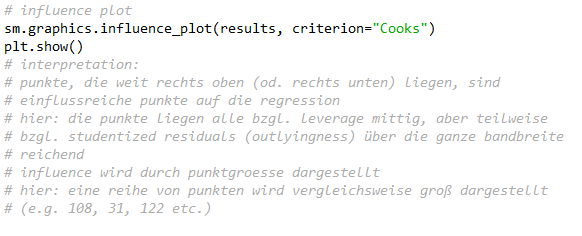
Der zweite Versuch ist **ein lineares Modell (OLS) u.a. unter Berücksichtigung der Interaktionen zwischen den Variablen** f1, f2, f3, f4. Die Interaktionen sind nicht signifikant. Die erklärte Varianz (R-Squared bzw. adj. R-Squared) liegt bei 15% bzw. 10%, d.h. sie konnte im Vergleich zum ersten linearen Modell erhöht werden.



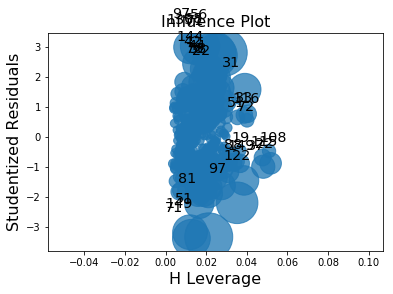


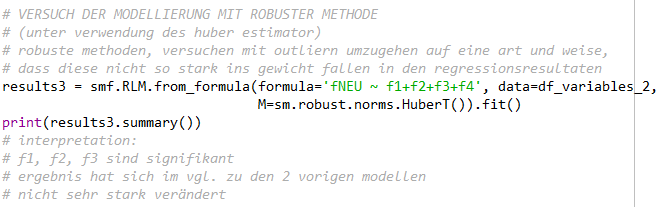
Zur **Regressionsdiagnose** des ersten linearen Modells wird ein Q-Q-Plot berechnet, der die theoretischen Quantile mit den empirischen Quantilen vergleicht. Wenn die Punkte direkt auf der Gerade liegen würden, wäre die Normalverteilungsannahme der Fehlerterme voll erfüllt. In diesem Fall liegen die Punkte im Intervall von ca. -2 bis 2 auf der Gerade, aber fransen außerhalb dieses Intervalls aus und weichen stärker von der Gerade ab. Die Normalverteilungsannahme der Fehlerterme ist also nicht voll erfüllt.



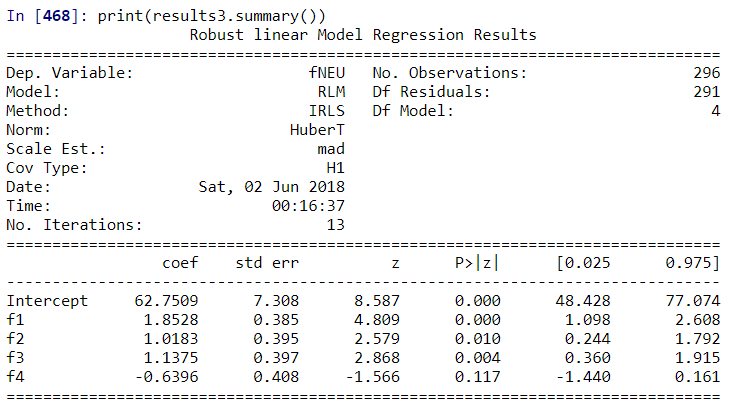


Für die Ergebnisse des ersten linearen Modells wurde ebenfalls ein Influence Plot erstellt. Dieser plottet Leverage und Residuals. „Einflussreiche“ Punkte, die Auswirkungen auf die Regressionsergebnisse haben, liegen rechts oben bzw. insgesamt rechts. In diesem Fall liegen die Punkte alle mittig bzgl. Leverage, aber teilweise bzgl. Studentized Residuals („outlyingness“) über die ganze Bandbreite reichend. Der Einfluss wird durch die Punktgröße dargestellt. Vergleichsweise groß werden hier z.B. die Punkte 97, 108, 19, 122 oder auch 72 dargestellt.

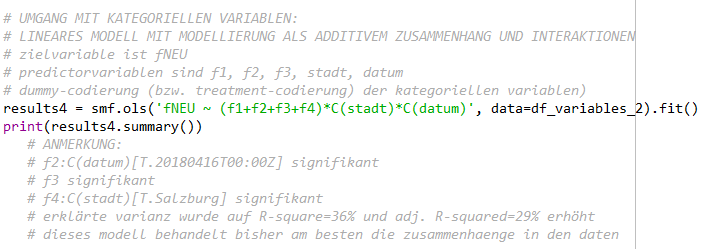




Der dritte Versuch berechnet ein **robustes lineares Modell nach dem Ansatz von Huber unter Betrachtung eines linearen Zusammenhangs zwischen den Variablen f1, f2, f3 und f4**. Robuste Methode versuchen mit Outliern umzugehen, indem sie diese nicht so stark ins Gewicht fallen lassen im Zuge der Regression. In der untenstehenden Zusammenfassung der Resultate sieht man, dass f1, f2 und f3 signifikant sind.



--------------------------------------------------------------------------------------------------------------



Die **kategoriellen Variablen** (d.h. Faktoren, die keine numerischen Werte habe, wie hier z.B. „stadt“ und „datum“) können **mittels Dummy-Codierung berücksichtigt** werden, wie es im vierten Modellversuch gezeigt wird. Es ergibt sich, dass durch diese Konstellation die erklärte Varianz (R-Squared bzw. adj. R-Squared) auf 29% bzw. 36% erhöht werden kann (siehe untenstehenden Output). Bezüglich der erklärten Varianz ist dieses bisher das beste Modell.

