# Aufgabe 1 in R

**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).

**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.

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

In dieser Dokumentation wird der lauffähige R-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 R-File, das ebenfalls auf Moodle hochgeladen wurde.

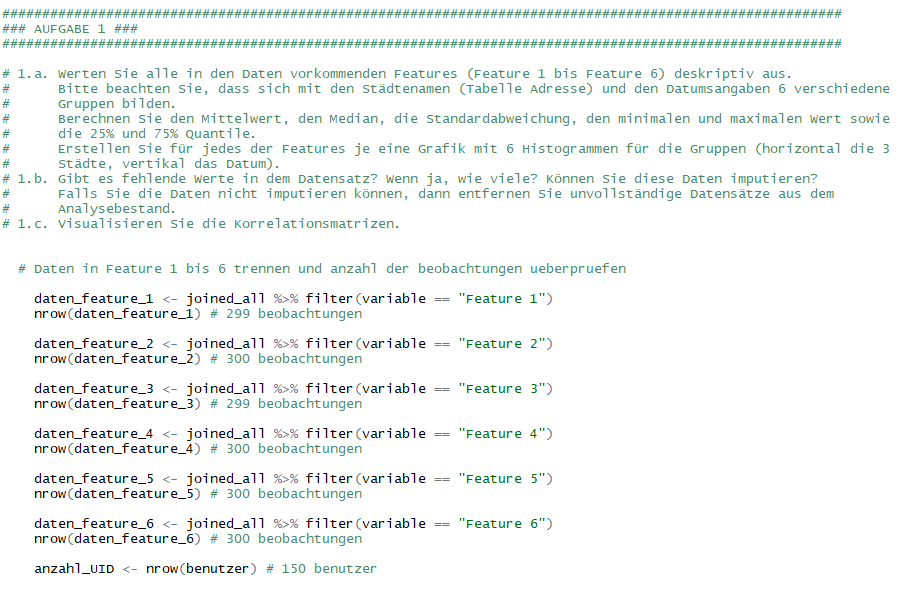
***Im Folgenden wir der R-Code erklärt und dokumentiert:***



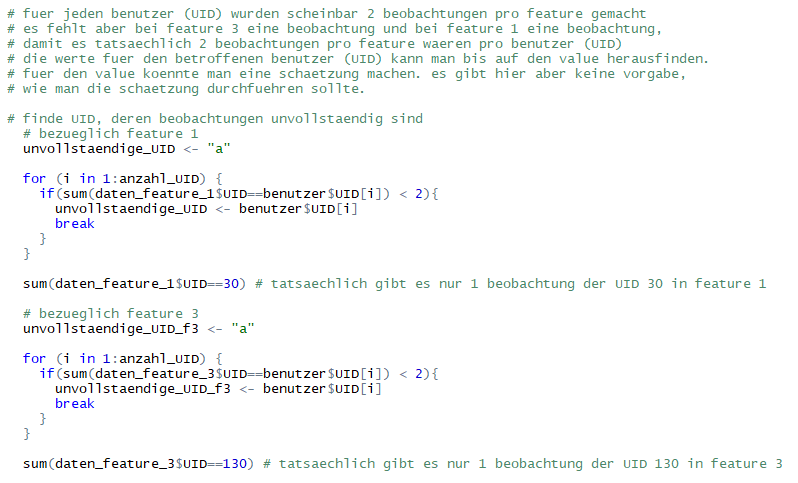
Zunächst wird sichergestellt, dass keine alten Objekte im File sind, und die notwendigen Libraries werden installiert bzw. geladen.



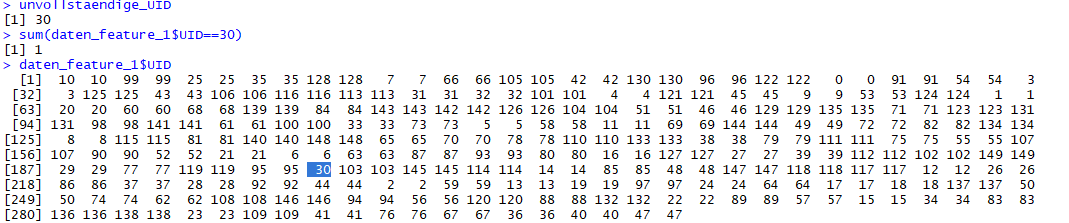
Die zur Bearbeitung notwendigen Daten werden aus den CSV-Files mittels read.csv-Funktion eingelesen. Da die gewonnenen Daten in Normalformen in drei Files vorliegen, müssen die Daten erst passend zusammengeführt werden mittels Joins. Die Dataframes „benutzer“ und „adressen“ werden über die Spalte „PLZ“ zusammengeführt. Das neu entstandene Dataframe „joined\_benutzer\_adressen“ wird über die Spalte „UID“ zusammengeführt mit dem Dataframe „features“.

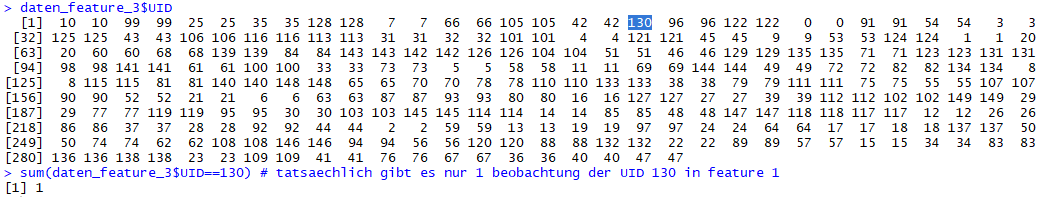


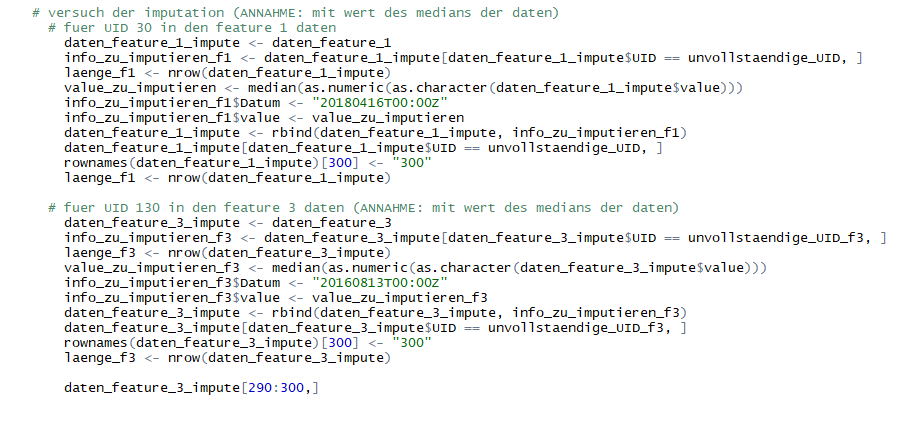
Die Daten werden nach Features (1 bis 6) getrennt, und die Anzahl der Beobachtungen pro Feature wird überprüft. Für jeden Benutzer (bzw. UID) wurden scheinbar 2 Beobachtungen pro Feature gemacht – also insgesamt pro Feature 300 Beobachtungen. Es fehlt daher bei Feature 3 und bei Feature jeweils 1 Beobachtung von einer UID. Diese betroffenen UIDs müssen ermittelt werden, oder vollständig entfernt werden aus den Daten, damit keine Probleme bei der weiteren Analyse entstehen.



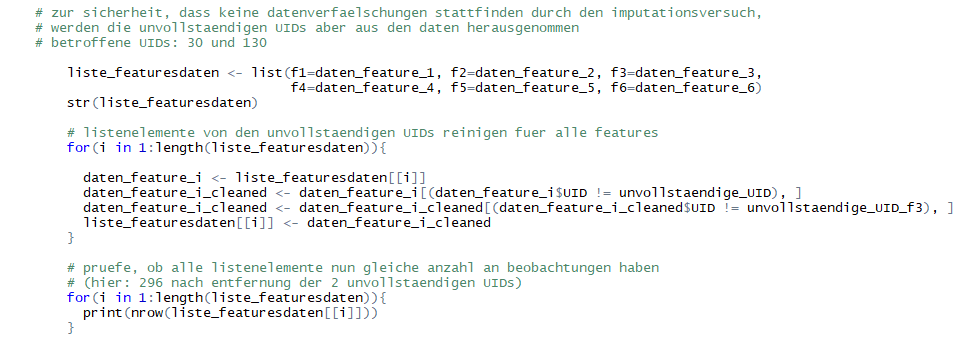
Die betroffenen UIDs müssen daher geeignet ermittelt werden, wie es im obigen Code-Stück dargestellt wird. Es zeigt sich, dass in den Daten zu Feature 1 eine Beobachtung von UID 30 fehlt, und im Feature 3 eine Beobachtung von UID 130, wie auch untenstehend ersichtlich ist:



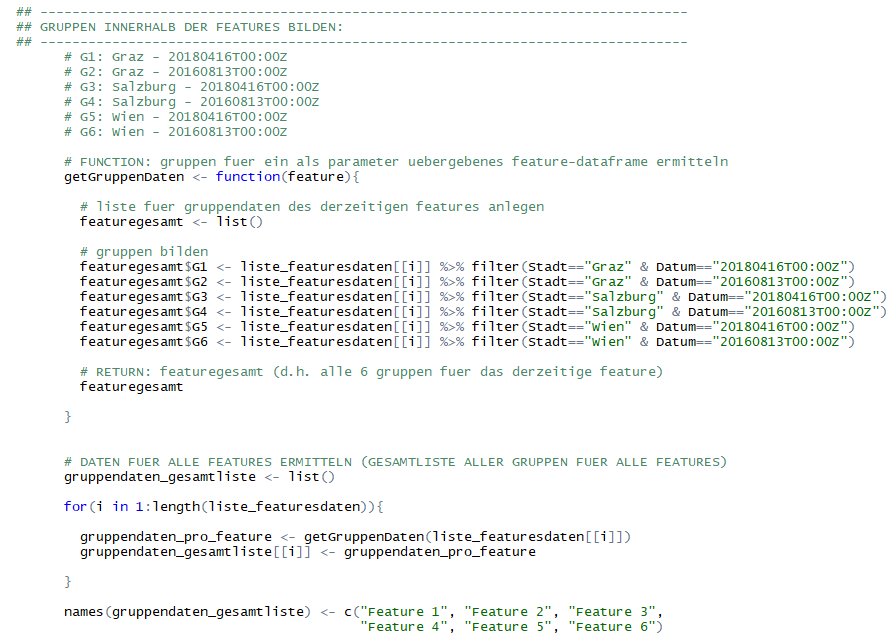




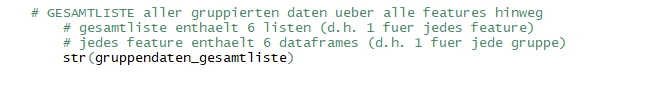
Alle Daten außer dem “value“ der jeweiligen Beobachtungen können unter Zuhilfenahme der vorhandenen Daten ermittelt werden. Der „value“ muss geschätzt werden oder nachdem hier die Daten relativ zentriert verteilt sind, kann auch der Median innerhalt der jeweiligen Featuredaten als Wert genommen werden, wie im Code implementiert.



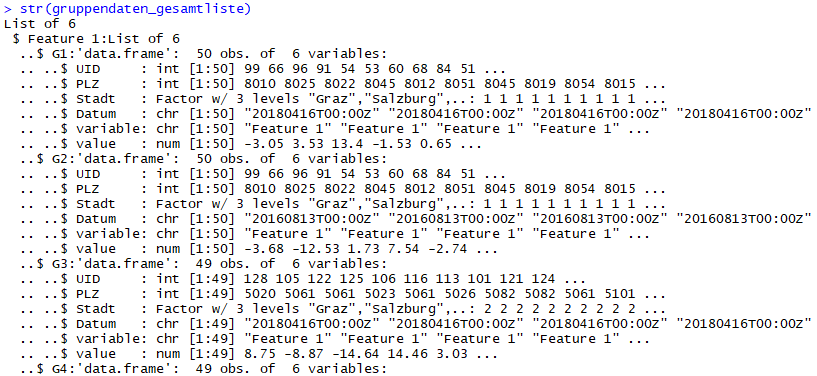
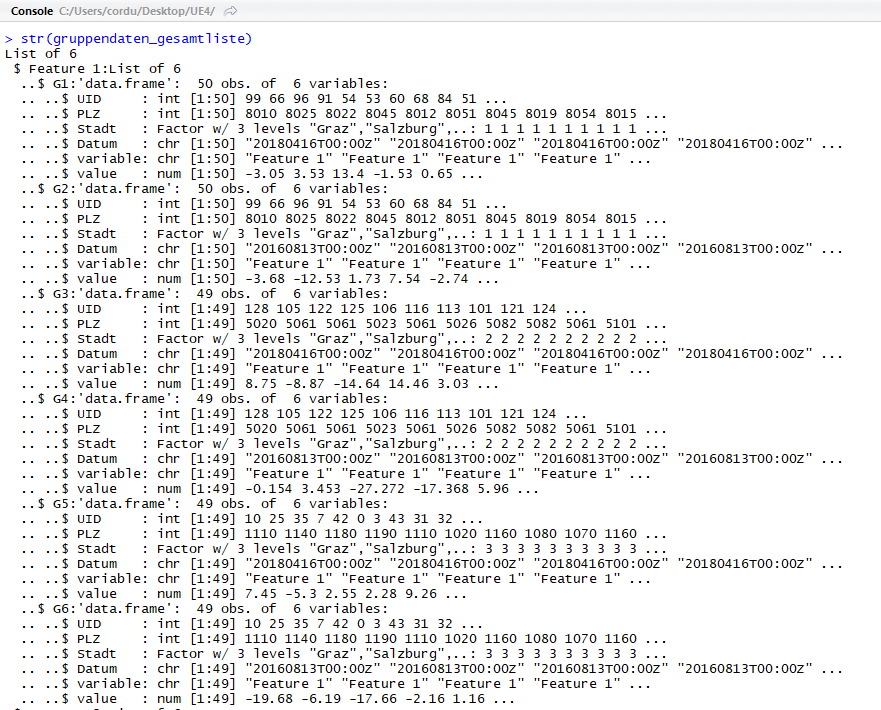
Da ich sicher gehen wollte, dass keine Datenverfälschung durch meine Schätzung bzw. Wertannahme ausgelöst werden, habe ich mich entschieden, den Imputationsversuch durchzuführen, um zu zeigen, dass dieser möglich ist, aber für die tatsächliche Analyse habe ich nun die betroffenen UIDs aus dem Datensatz herausgenommen. Es wird also rein auf Basis von vollständigen Beobachtungen analysiert.

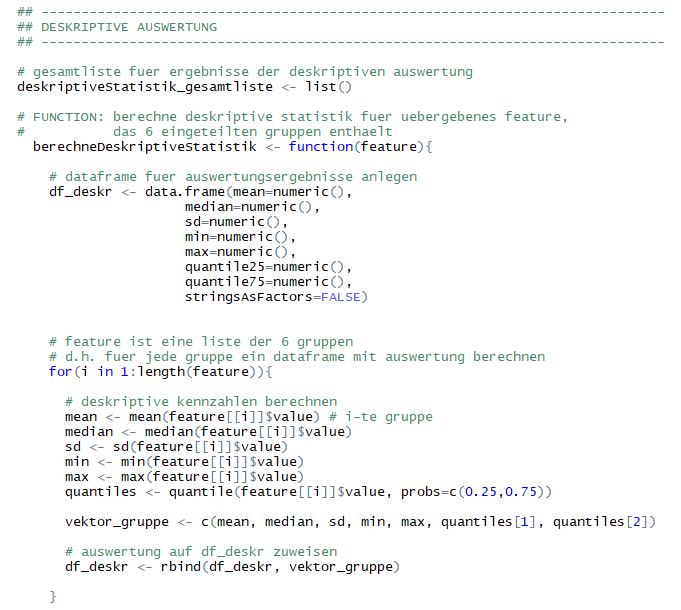


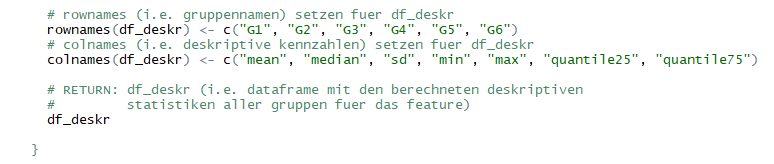
Die Bildung von Gruppen erfolgt gemäß den Kombinationen von „stadt“ und „datum“, die ich in G1 bis G6 eingeteilt habe. Die Funktion „getGruppenDaten“ nimmt als Parameter „feature“, also die Daten eines Features, und bildet dafür die jeweils 6 Gruppen. Diese Funktion wird dann für alle Daten aus der gesamten Featuresdatenliste aufgerufen.



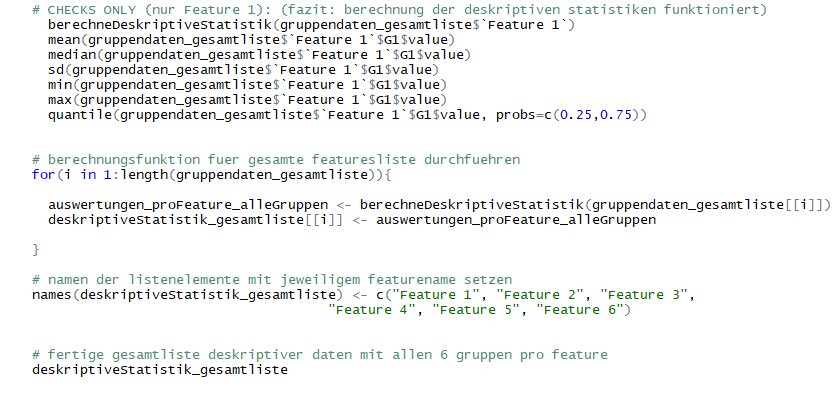
Das Ergebnis der Einteilung in Gruppen ist das Objekt „gruppendaten\_gesamtliste“, das eine Liste bestehend aus den 6 Features ist. Jedes Feature hat in sich 6 Dataframes, d.h. jeweils ein Dataframe pro Gruppe. Die folgende Abbildung zeigt die Struktur davon:



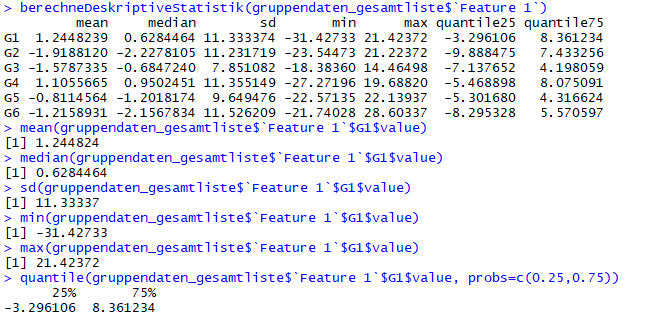




Für die deskriptive Auswertung des Datensatzes dient die Funktion „berechneDeskriptiveStatistik“, die ein feature als Parameter übergeben bekommt. Sie berechnet pro Feature für jede Untergruppe die gefragten deskriptiven Statistikkennzahlen (mean, median, sd, min, max, quantile25, quantile75). Der Rückgabewert der Funktion ist das Dataframe „df\_deskr“, das alle berechneten deskriptiven Statistiken aller Gruppen für das übergebene Feature berechnet.



Die Berechnungsfunktion wird für die gesamte Featuresliste (und innerhalb pro Gruppe) ausgeführt. Dies ergibt schließlich die Gesamtliste aller deskriptiven Statistiken. Das Ergebnis sieht z.B. für das Feature 1 in der gruppen\_gesamtliste folgendermaßen aus, wobei die unteren Statements zur Überprüfung ausgeführt wurden, ob die Daten tatsächlich übereinstimmen:



Für den Gesamtdatenbestand ergeben sich die folgenden Auswertungen:

*Anmerkung zur Gruppenbenennung:*

*# G1: Graz - 20180416T00:00Z*

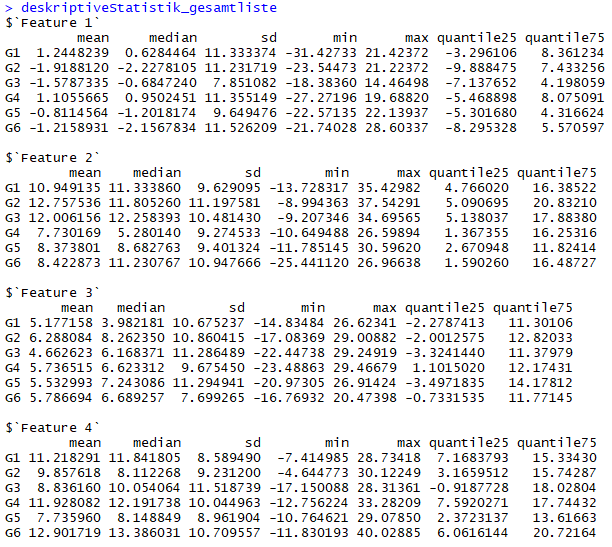
*# G2: Graz - 20160813T00:00Z*

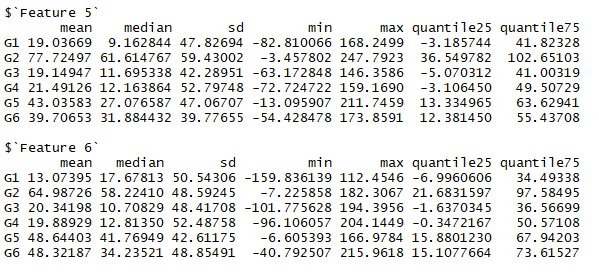
*# G3: Salzburg - 20180416T00:00Z*

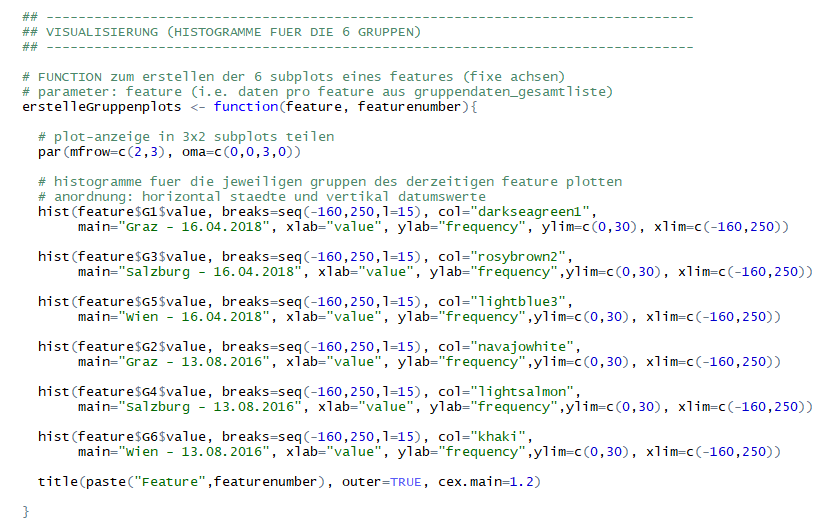
*# G4: Salzburg - 20160813T00:00Z*

*# G5: Wien - 20180416T00:00Z*

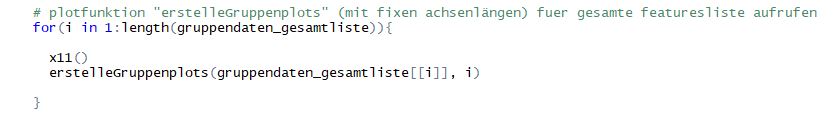
*# G6: Wien - 20160813T00:00Z*



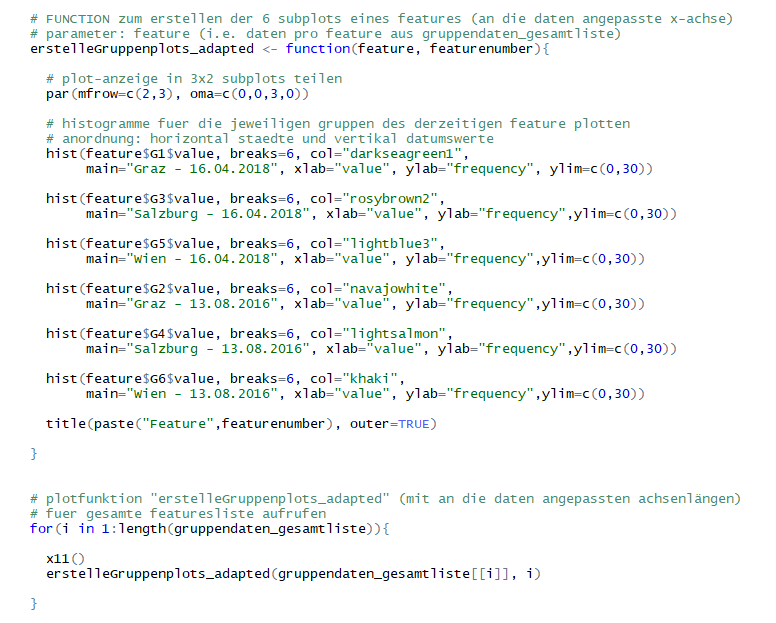




Die Visualisierung der Histogramme für die 6 Gruppen pro Feature erfolgt mit der Funktion „erstelleGruppenplots“, die die Parameter „feature“ und „featurenumber“ erfordert. Für jede Gruppe innerhalb des Features wird ein Histogramm in passender Farbe angelegt und der gesamte Plot ist in 3x2 (d.h. Städte x Datumswerte) Subplots eingeteilt. Diese Funktion stellt die Daten basierend auf fixen Achsenwerten dar, wodurch die Daten untereinander leichter vergleichbar sind.



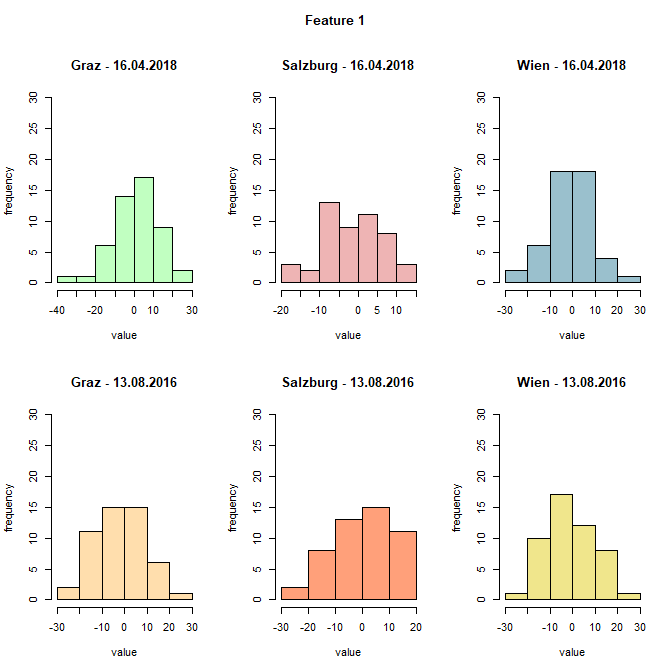
Die Plot-Funktion wird für die gruppendaten\_gesamtliste aufgerufen, wodurch für alle Gruppen die passenden Plots pro Feature erstellt werden.

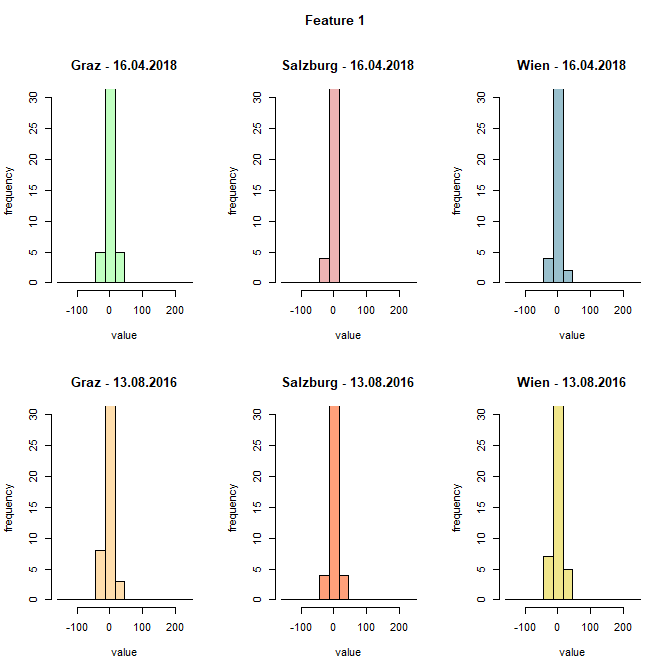


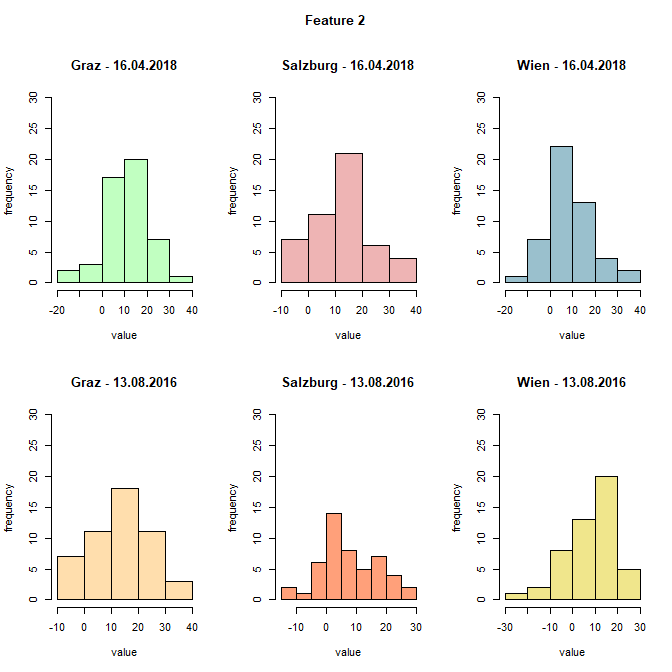
Die Funktion „erstelleGruppenplots\_adapted“ erfüllt einen ähnlichen Zweck wie die vorherige Funktion. Sie stellt die Daten allerdings basierend auf an die Daten angepasste x-Achsenlängen dar, wodurch ein genauerer Blick auf die Daten gegeben werden kann.

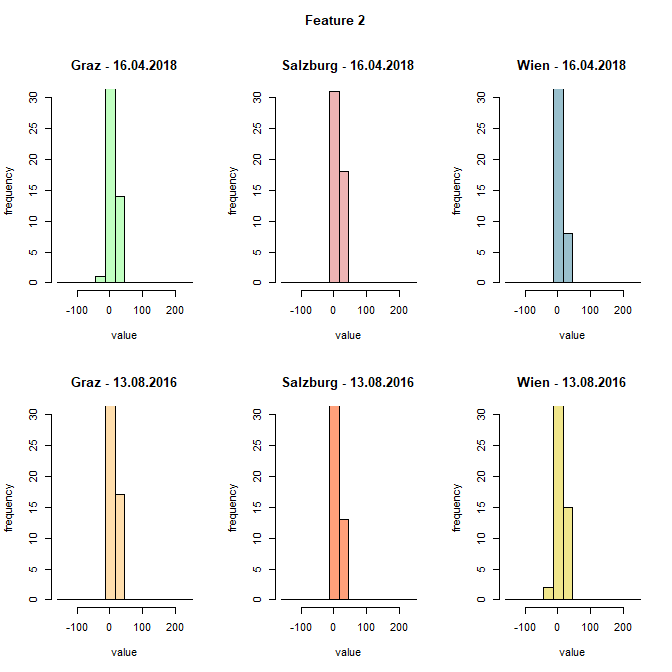
***Die Subplots (Histogramme) pro Feature:***

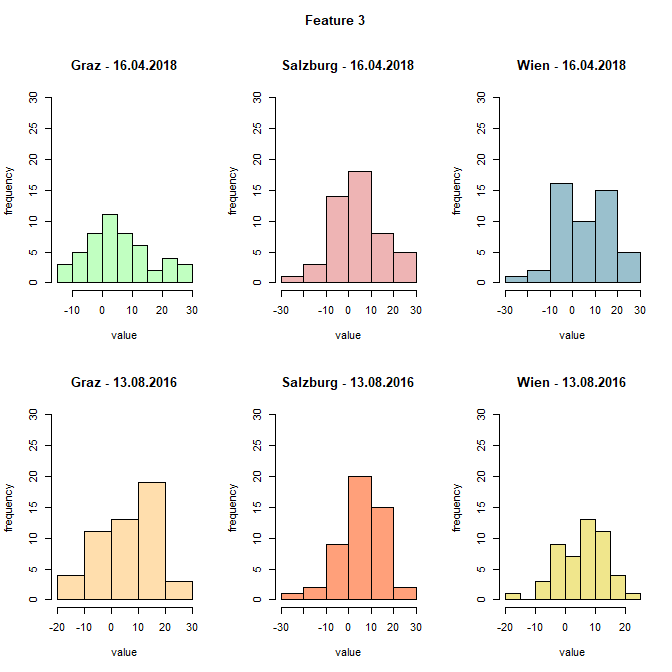
1. Output der Funktion „erstelleGruppenplots\_adapted“ (Darstellung auf an die Daten angepasster x-Achsenlänge)
2. Output der Funktion „erstelleGruppenplots“ (Darstellung auf fixer Achseneinstellung)

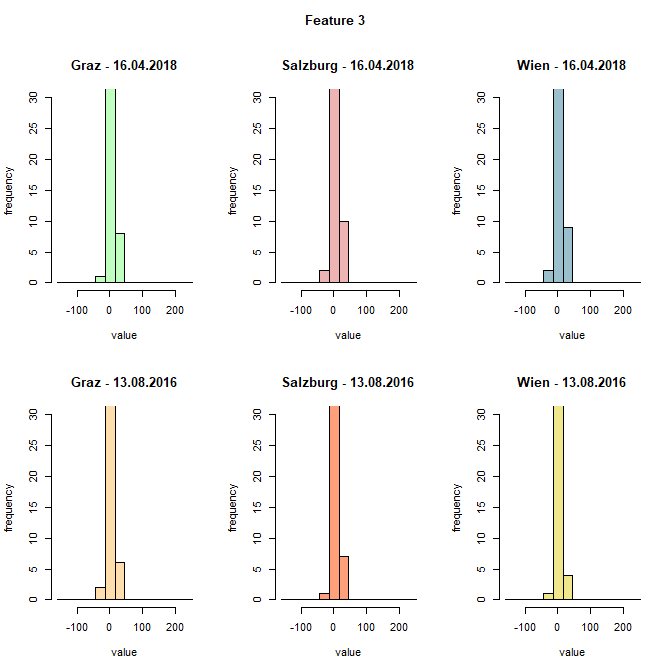


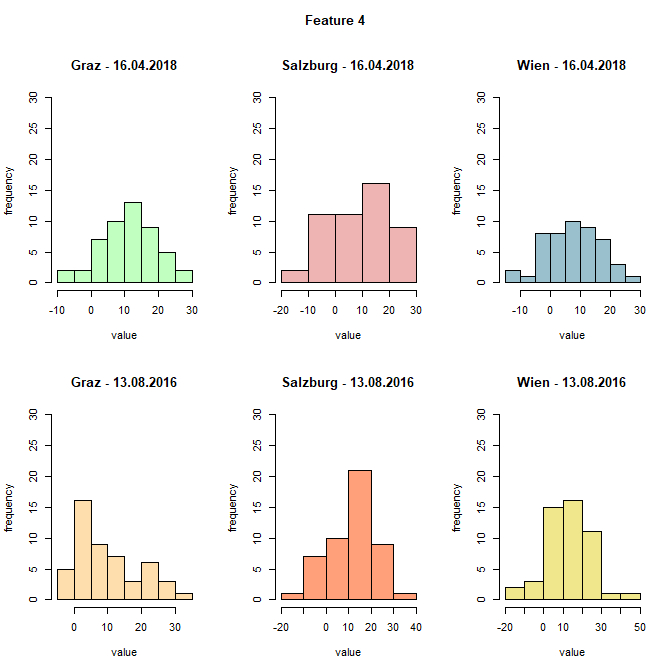


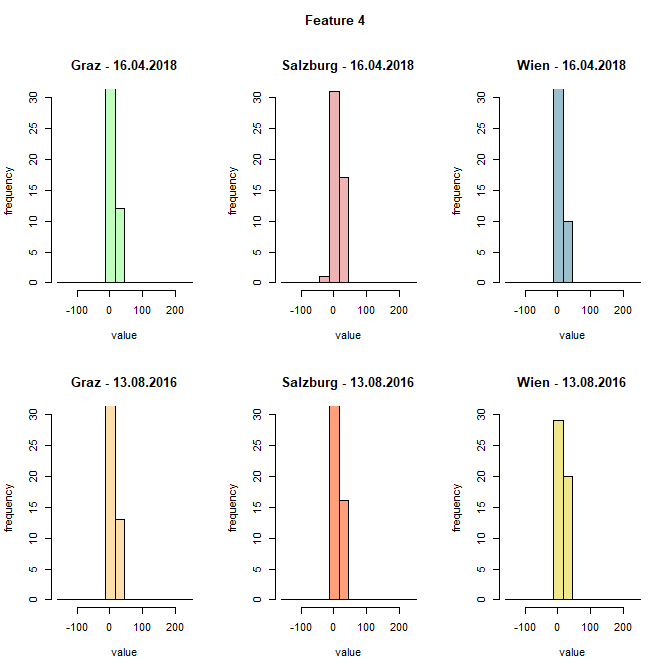


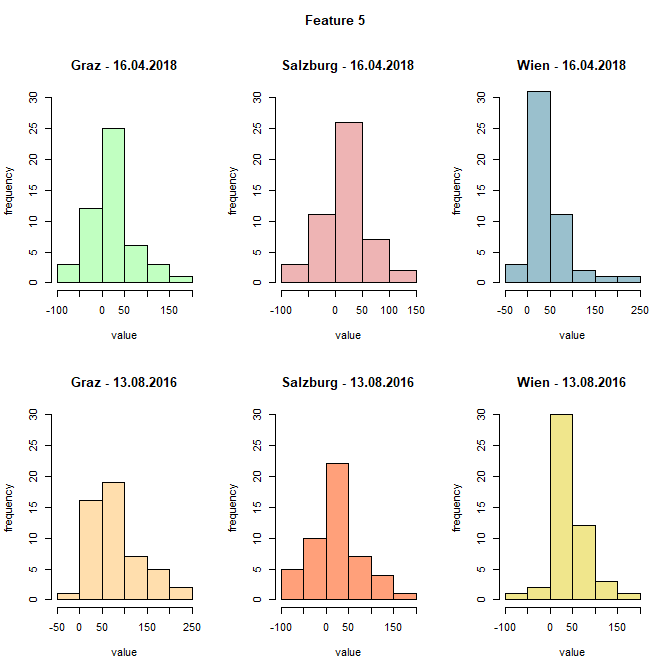


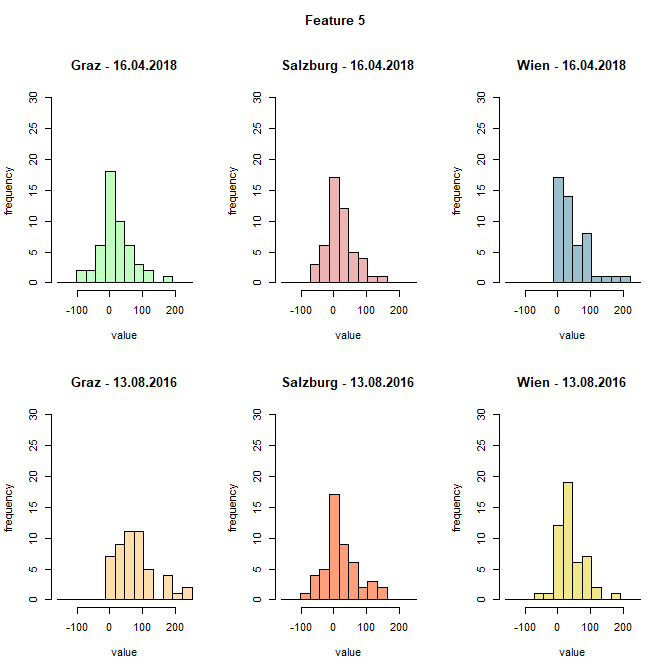


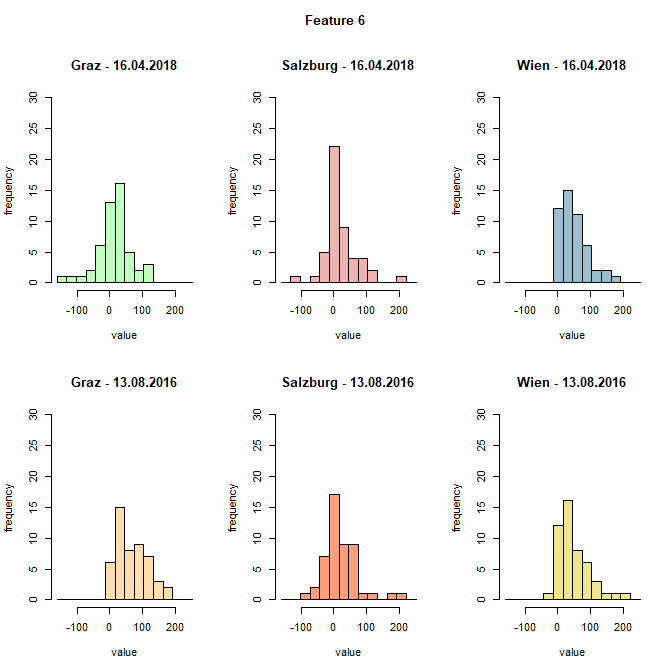
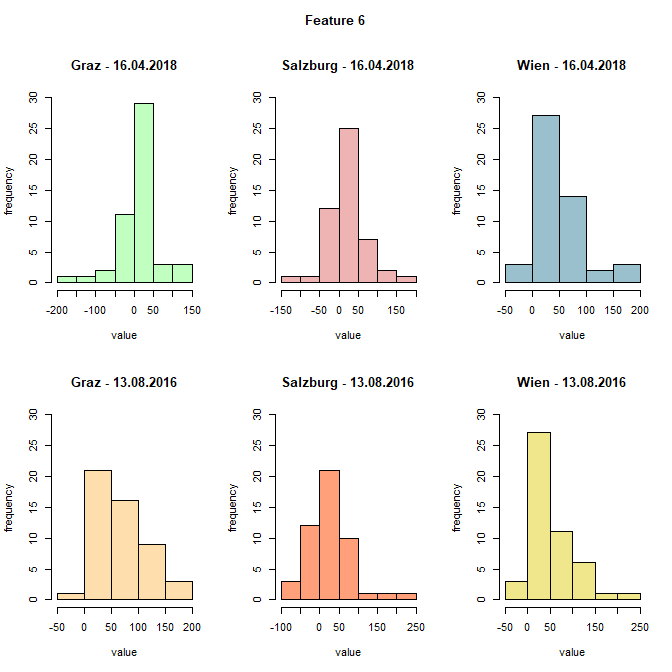


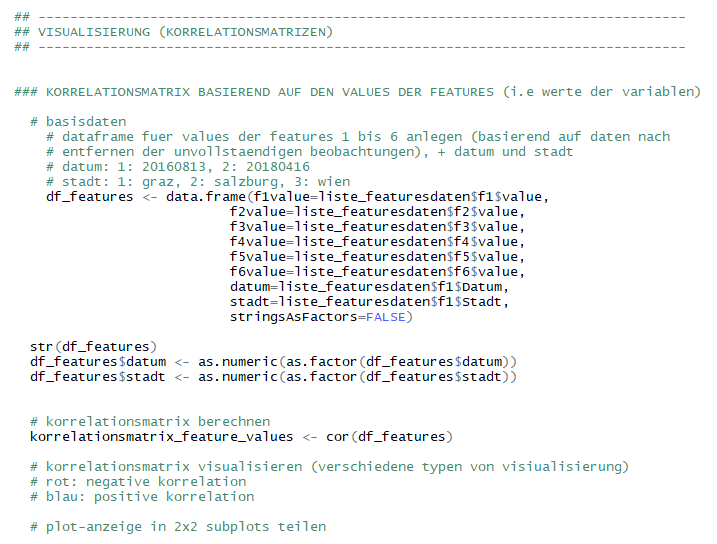




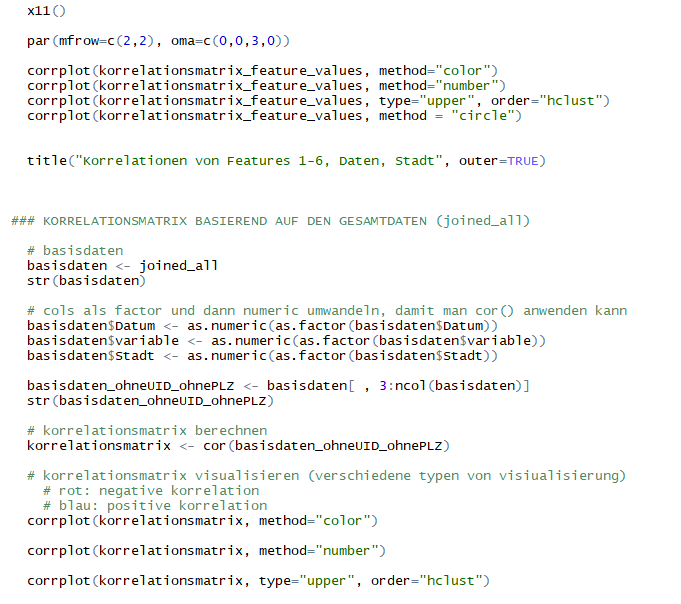


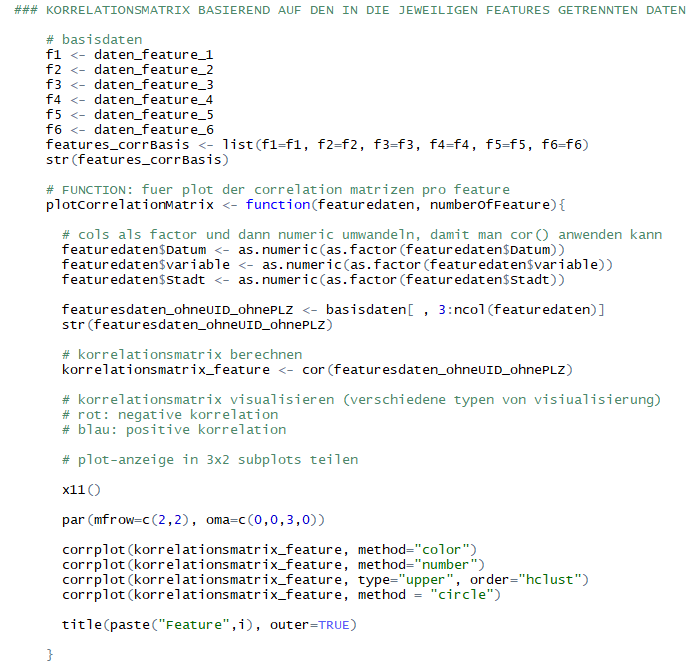


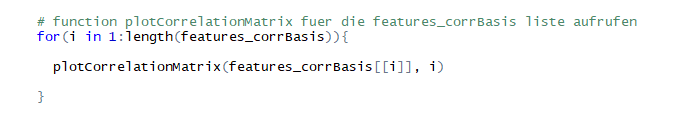




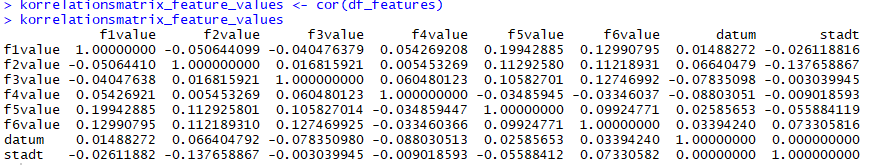
Zur Visualisierung der Korrelationsdaten werden die Daten für die jeweiligen Features sowie Datum und Stadt verwendet. Die Korrelationsmatrix wird mittels der im R schon vorhandenen Funktion „corr“ berechnet. Für die Visualisierung wird die Library „corrplot“ verwendet und daraus die Funktion „corrplot“. Letztere wird in verschiedenen Versionen basierend auf denselben Daten geplottet.

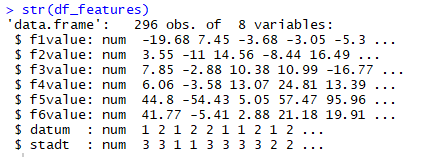






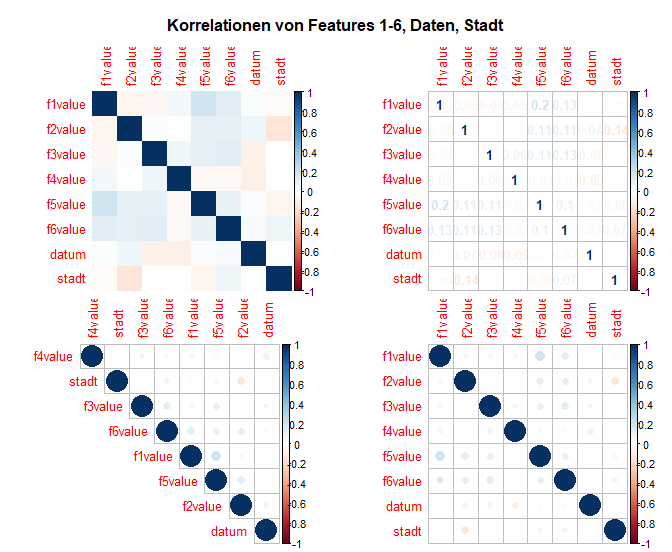
***Berechnete Korrelationsmatrix:***



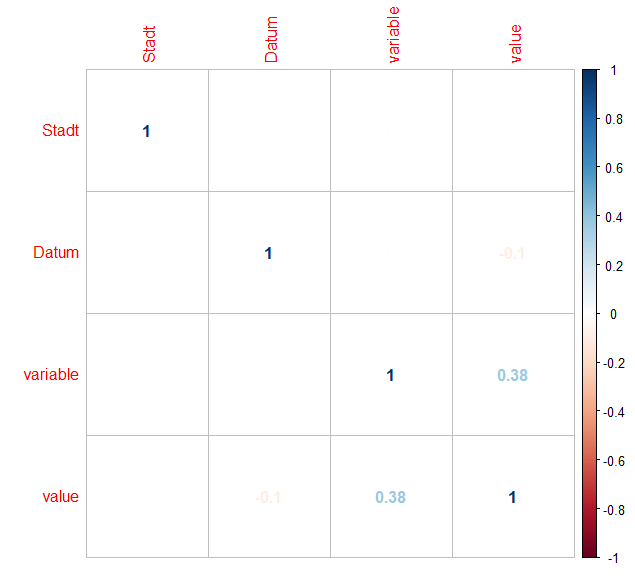
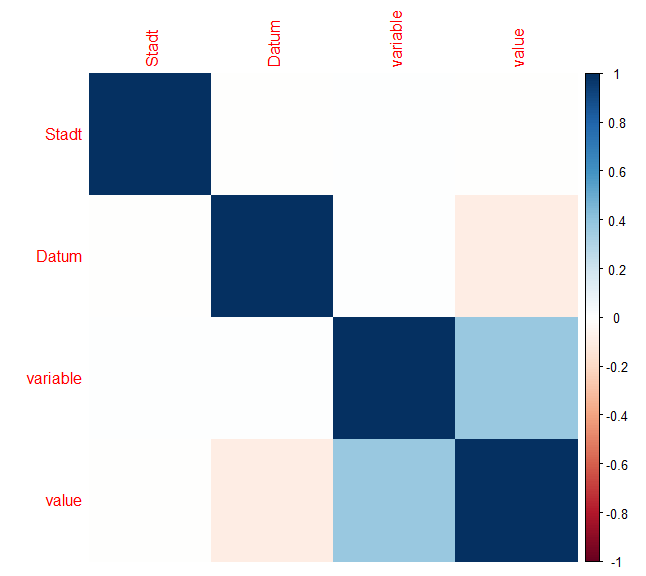


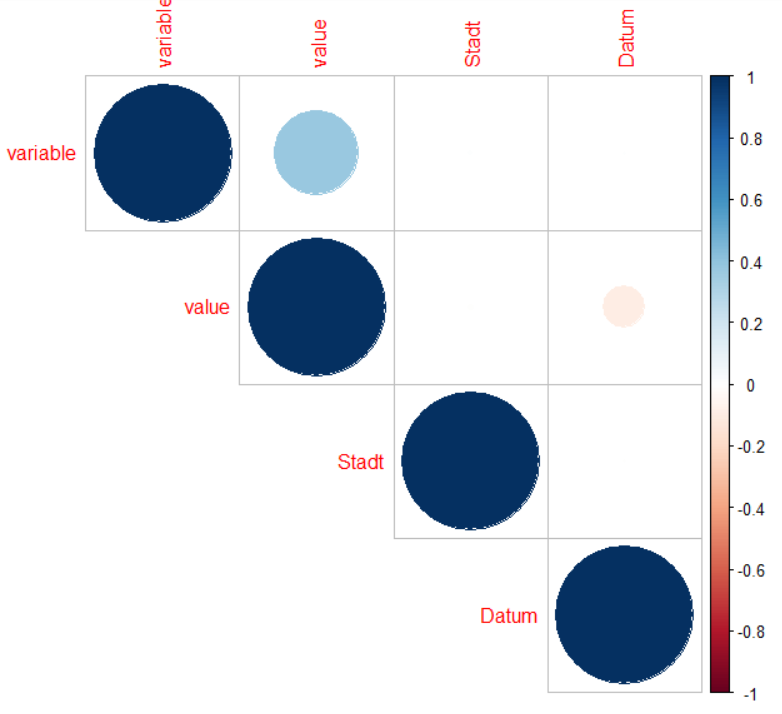
***Visualisierung der Korrelationen:***

* Rot: negative Korrelation
* Blau: positive Korrelation

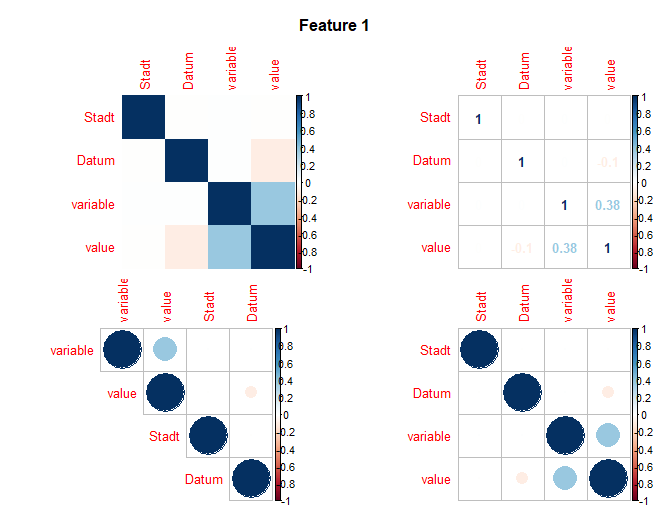


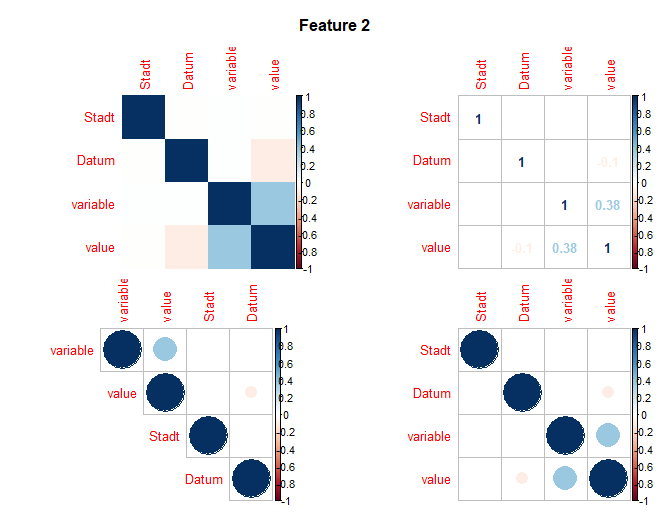
Korrelationen zwischen Stadt, Datum, variable und value:

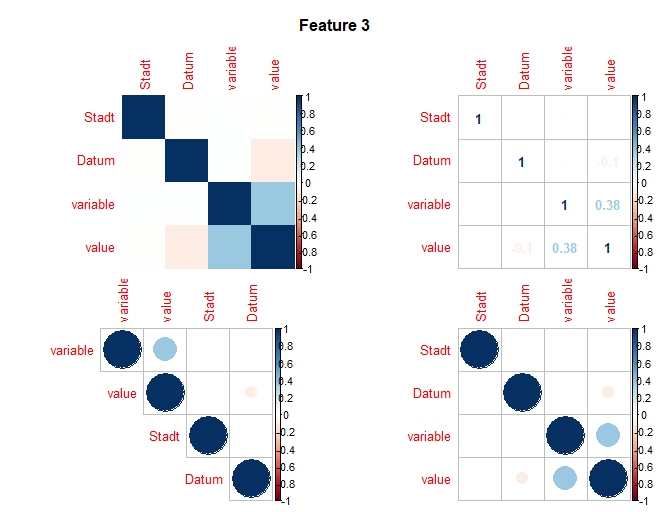


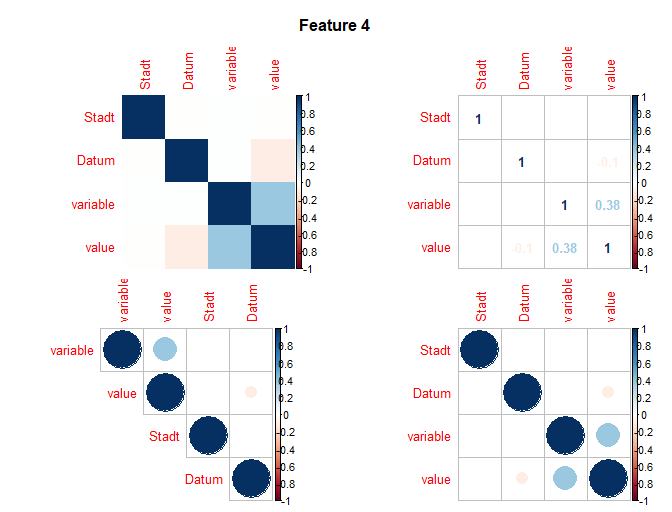


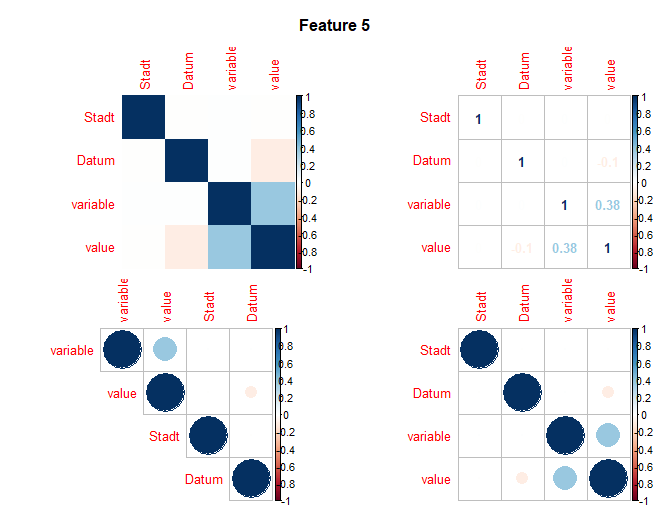
Korrelationen pro Feature (UID und PLZ Spalte weggelassen):

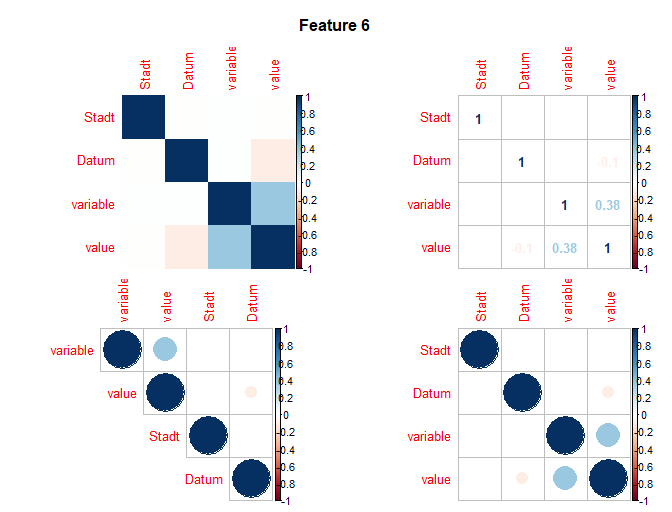




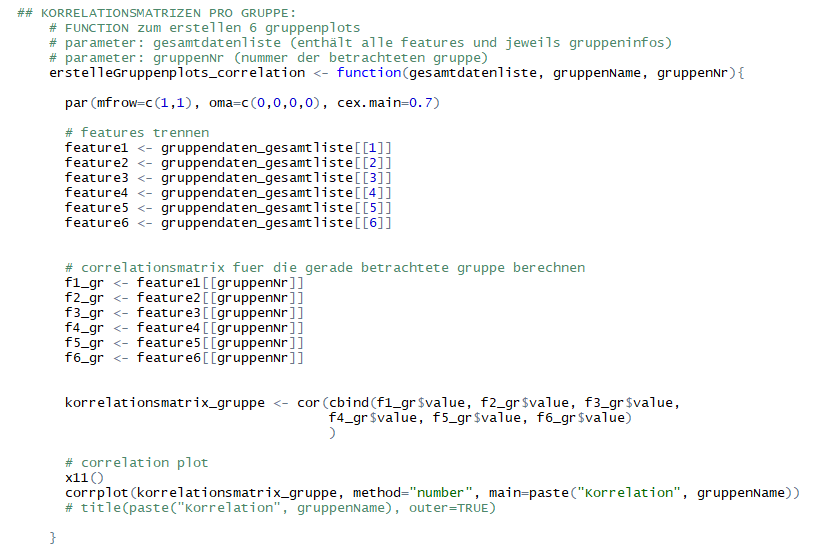


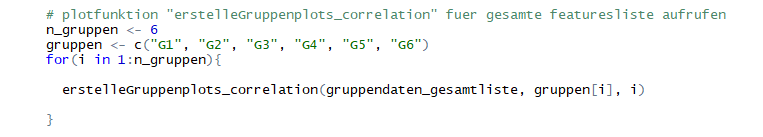






**Korrelation basierend auf den Gruppen pro jeweiligem Feature:**





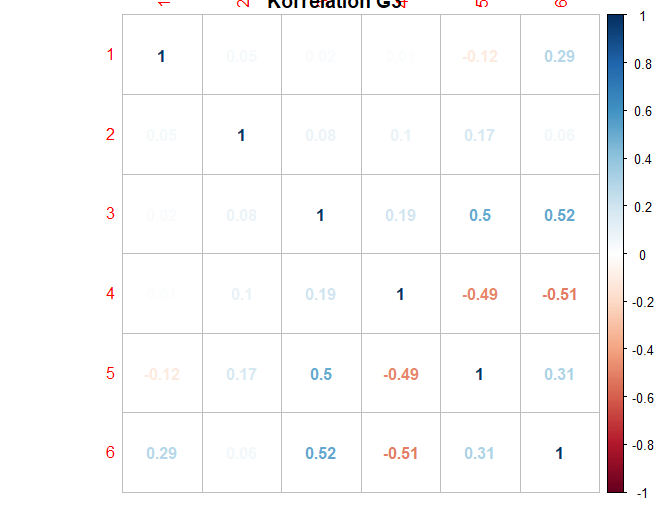
**G1:**



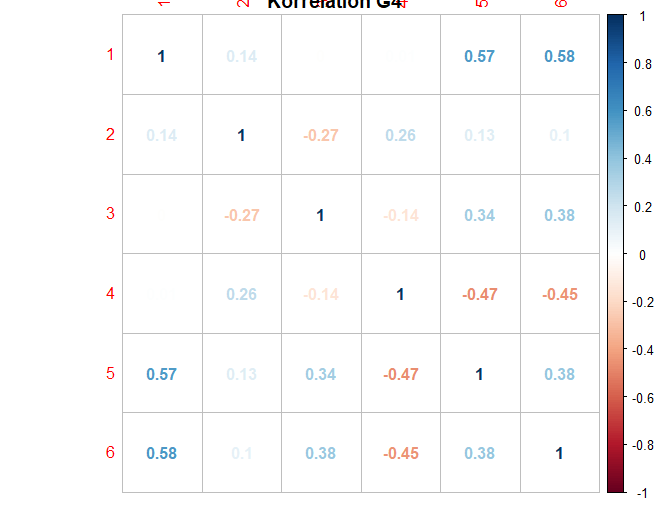
**G2:**



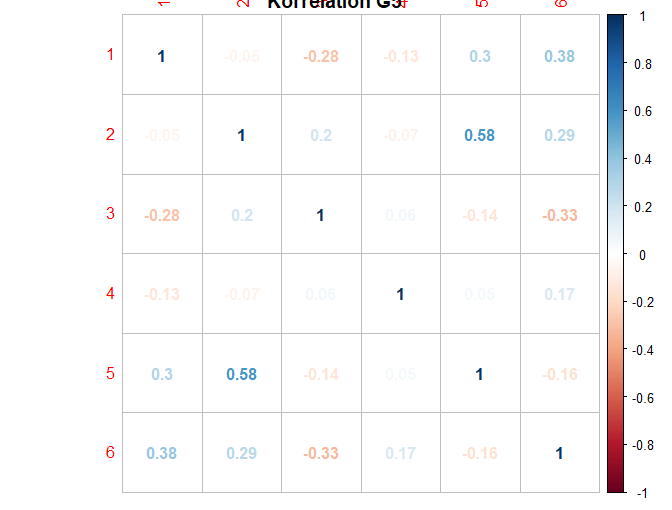
**G3:**



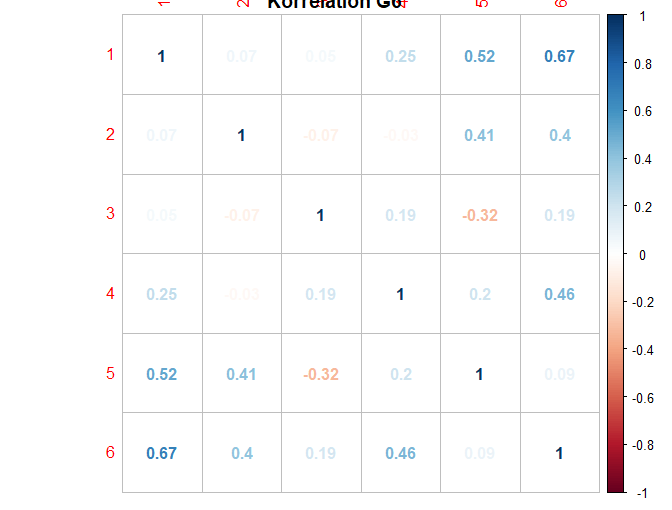
**G4:**



**G5:**



**G6:**



Anmerkung:

Man sieht aus der Visualisierung der Korrelationsmatrizen der Feature-Werte innerhalb der 6 Gruppen, dass die Korrelationen der verschiedenen Variablen je nach Gruppen variieren.

# Aufgabe 2 in R

**Aufgabe 2 – Aufgabenstellung:**

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

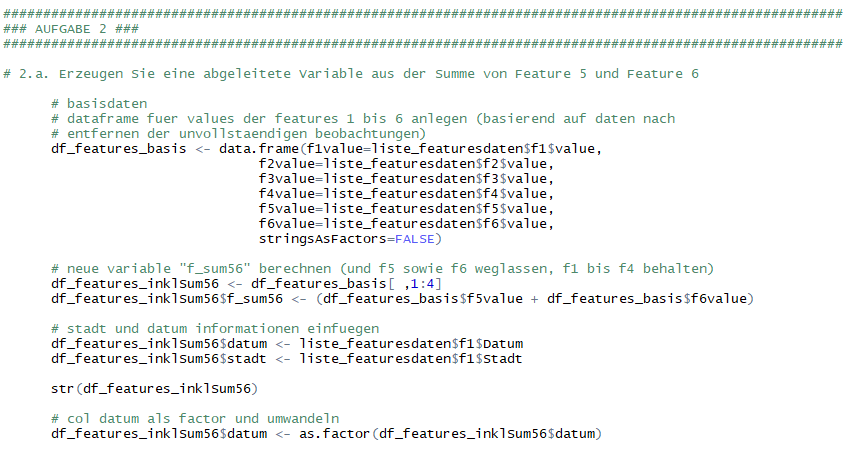
**b.** Gibt es Korrelationen zwischen den verbleibenden Variablen und der neuen abgeleiteten Variable?

**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?

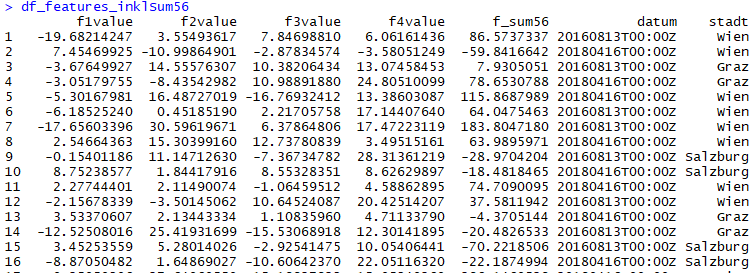
**e.** Beschreiben Sie Ihre Modellierungsergebnisse und erzeugen Sie Grafiken um Ihre Ergebnisse zu dokumentieren.

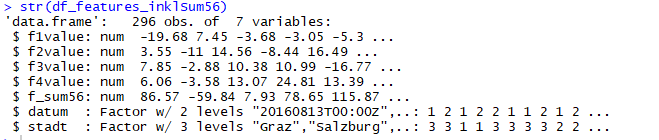
**f.** Welche Modellierungsmethode verwenden Sie und warum haben Sie sich für dieses Modell entschieden?

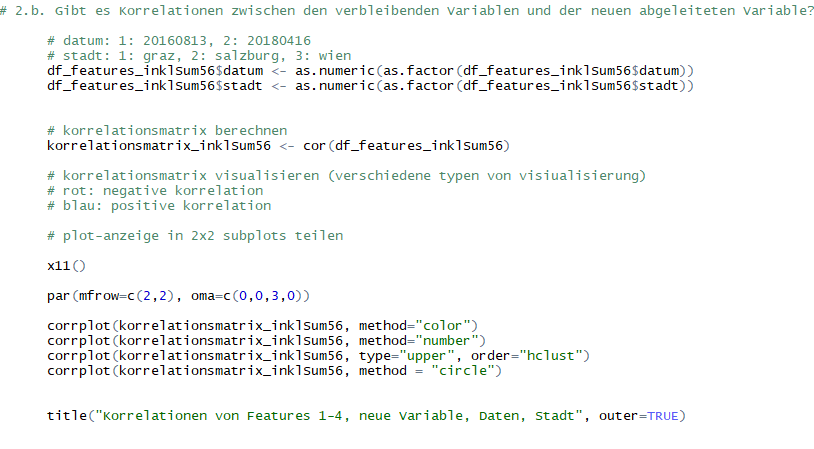


Aus den Summen von Feature 5 und Feature 6 wird in den gesamten Featuredaten eine neue Variable erstellt und an das Dataframe angehängt. Die Features 5 und 6 werden weggelassen. Auch Datum und Stadt werden hinzugenommen.

Das neu entstandene Dataframe sieht folgendermaßen aus (siehe untenstehender Ausschnitt):

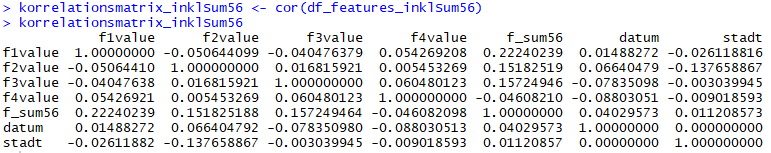




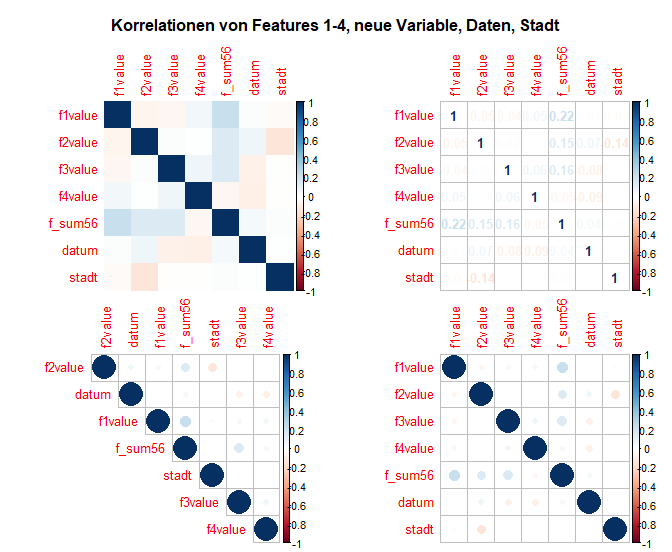


Für die Untersuchung der Korrelation wird die Korrelationsmatrix für das neue Dataframe df\_features\_inklSum56 gebildet. Basierend darauf wird corrplot aufgerufen, und die Korrelation wird auf verschiedene Arten visualisiert.

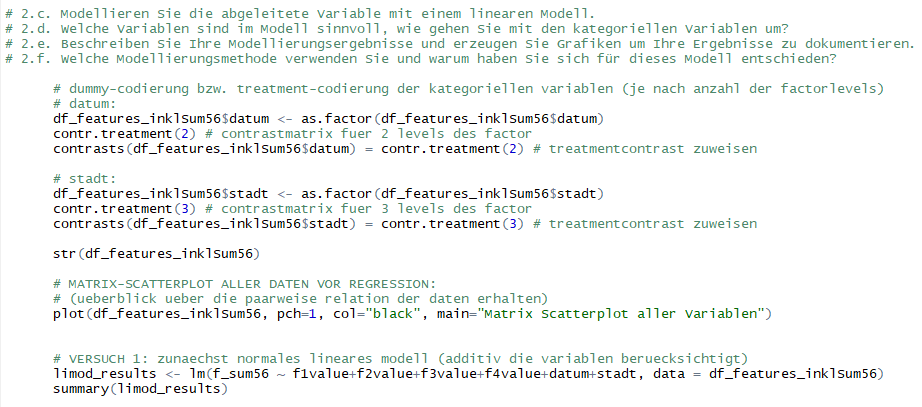
Die Korrelationsmatrix:



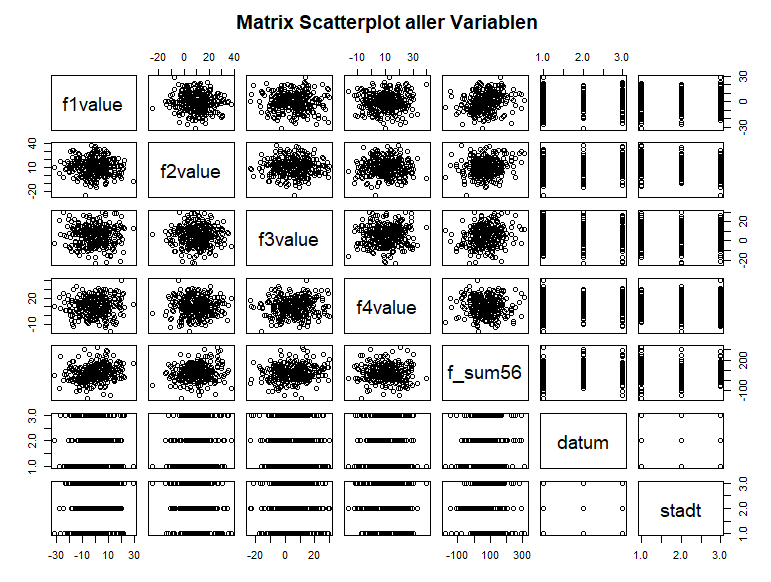
Die Visualisierung der Korrelation:

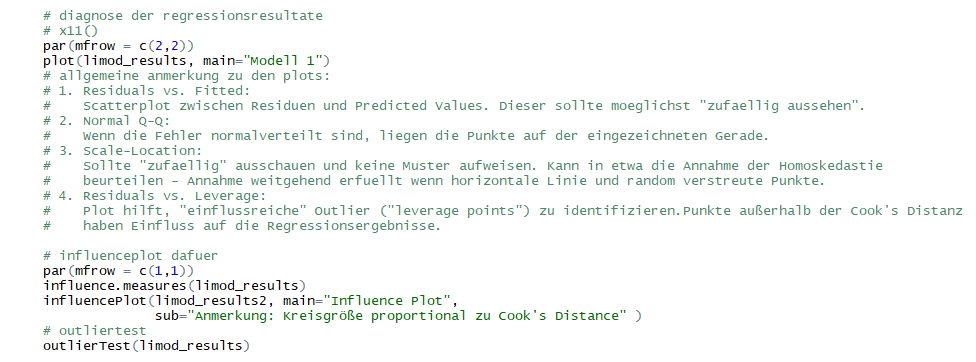


Lineares Modell – Versuch 1 (additiver Zusammenhang):

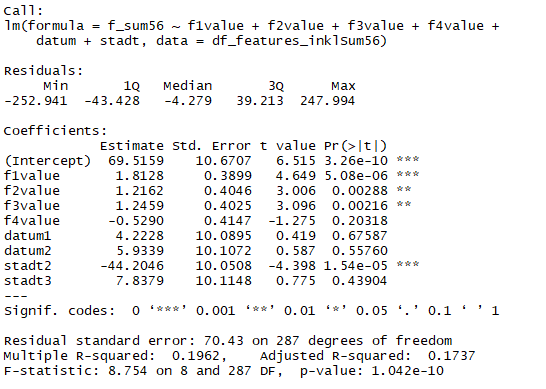


Das Datum und die Stadt werden zunächst als Factor verwendet und mittels Treatment-Codierung werden die Kontraste gesetzt. Danach werden Versuche von linearen Modellen gemacht. Ein erster Matrix-Scatterplot der Daten gibt einen Überblick über die Datenlage, wie untenstehend zu sehen ist:

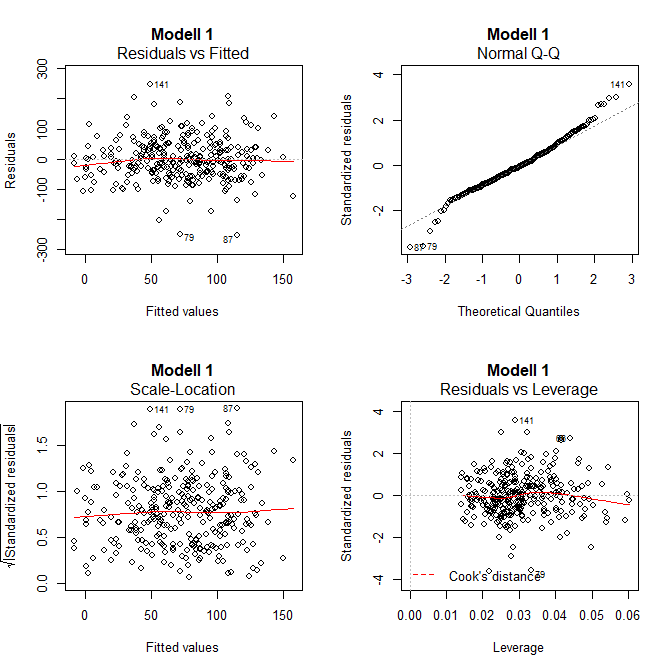




Mit der Funktion lm() wurde ein erster Versuch für ein lineares Modell gemacht, in dem die Regressoren additiv betrachtet wurden. Die adj. erklärte Varianz (adj. R-Squared) liegt bei 17.37%. Signifikante Ergebnisse liegen bei f1value und stadt2 (i.e. Gruppe Salzburg) vor. Die Werte für f2value und f3value sind ebenfalls signifikant. Bei Datum, f4value und stadt3 (i.e. Wien) liegen keine signifikanten Ergebnisse vor. Die gesamte Auswertung des linearen Modells sieht folgendermaßen aus:



Zur Diagnose der Regression wurden Residuals vs. Fitted Values, Normal Q-Q Plot, Scale-Location Plot und Residuals vs. Leverage betrachtet:



Residuals vs. Fitted zeigt einen Scatterplot zwischen Residuen und Predicted Values. Dieser sollte möglichst „zufällig“ aussehen. Potenziell kann man in einem solchen Plot erkennen, ob ein nicht-linearer Zusammenhang besteht.

Im vorliegenden Fall sieht der Plot „zufällig“ aus und es ist kein nicht linearer ersichtlich. Die Wahl der linearen Modellierung ist also in Ordnung.  
Der Normal Q-Q Plot dient zu Analyse, ob die Fehler normalverteilt sind. Wenn die Punkte auf der eingezeichneten Gerade liegen, sind die Fehler normalverteilt. Falls nicht, ist die Annahme der normalverteilten Fehler nicht erfüllt.

Im vorliegenden Fall liegen die liegen die Punkte in der Mitte direkt auf der Gerade, aber an den Enden weichen sie stärker davon ab. Die Normalverteilungsannahme der Fehler ist daher nicht voll erfüllt.

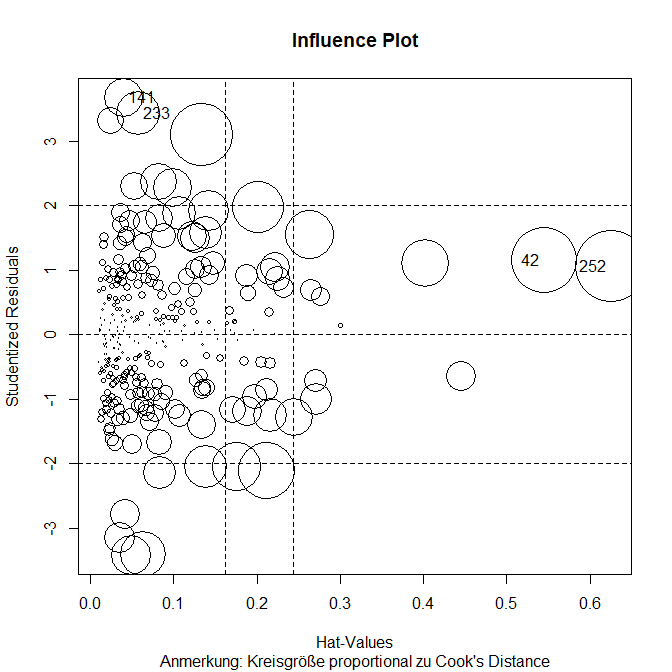
Der Scale-Location Plot sollte möglichst „zufällig“ ausschauen und keine Muster aufweisen. Damit kann man in etwa die Annahme der Homoskedastizität (Varianzhomogenität bzgl. Fehlertermen) beurteilen. Die Annahme ist weitgehend erfüllt, wenn der Plot eine annähernd horizontale Linie und „zufällig“ verstreute Punkte aufweist.

Der Plot sieht im obigen Modell „zufällig“ aus und es gibt keine starken einseitigen Abweichungen und Muster im Plot.

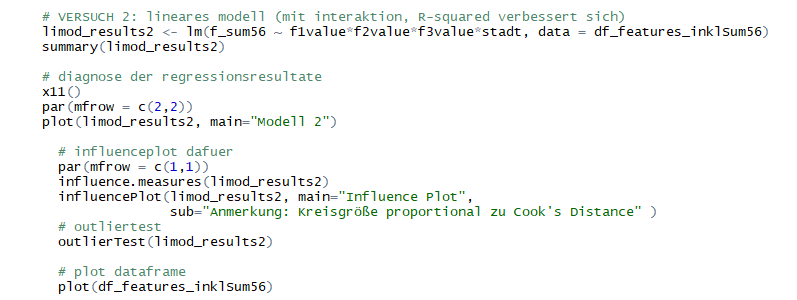
Der Residuals vs. Leverage Plot hilft, “einflussreiche Outlier“ (leverage points) zu identifizieren. Diese liegen außerhalb der Cook’s Distanz und haben Einfluss auf die Regressionsergebnisse.

Hier sind keine Punkte außerhalb der Cooks Distance Linie ersichtlich. Daher kann davon ausgehen, dass keine starke Beeinträchtigung der Regression durch einzelne influence points stattfindet. Punkt 141 und die umliegenden Punkte sollte jedoch genauer angesehen werden.

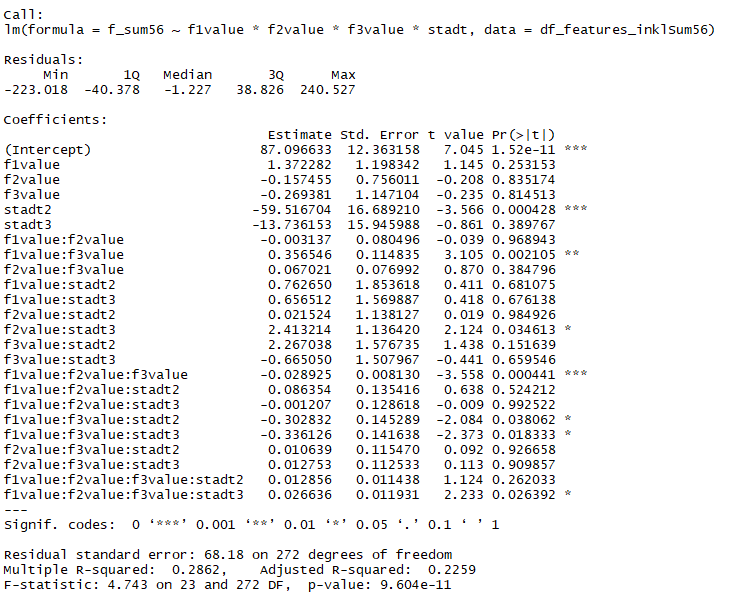
Untenstehend wird der Influence Plot gezeigt, der den Einfluss der Beobachtungen veranschaulicht. Die Beobachtungen 141, 233, 42 und 252 scheinen unter anderem bezüglich Influence größere Bedeutung zu haben.

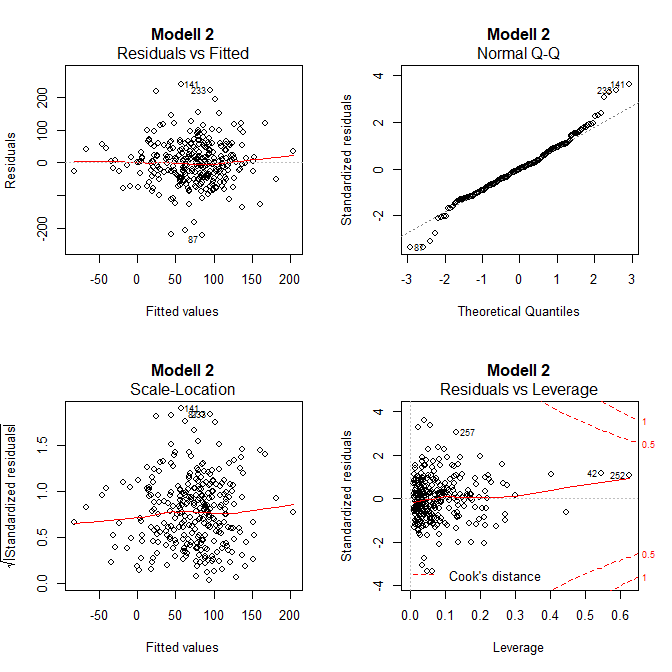


Lineares Modell – Versuch 2 (Zusammenhang mit Interaktion):



Im Versuch 2 wird ein lineares Modell mit Interaktion ausgewertet. Der Wert von adjusted R-Squared verbessert sich auf 22.59%. Die Interaktion von f1value:f2value:f3value und von f1value:f3value sind signifikant. Leicht signifikant sind auch die Interaktion von f2value:stadt3, f1value:f3value:stadt2, f1value:f3value:stadt3 und f1value:f2value:f3value:stadt3.





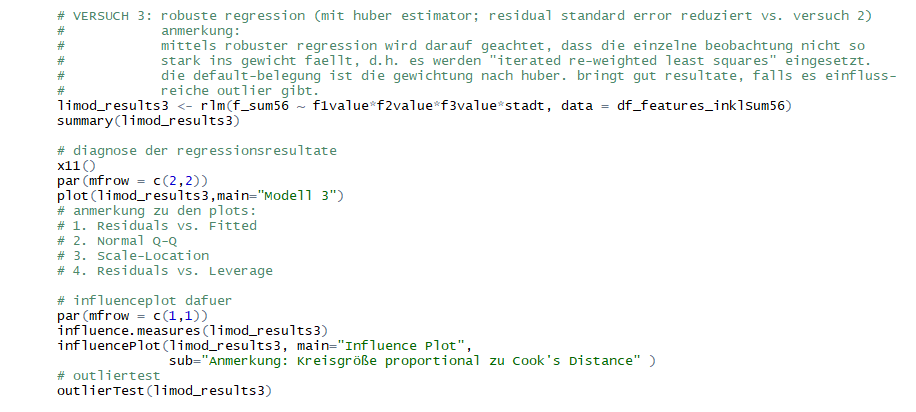
Hinsichtlich Residuals vs. Fitted Values ist im Vergleich zum ersten Modell eine stärkere Dichte der Punkte in der Mitte zu verzeichnen. Ein nichtlinearer Zusammenhang ist nicht direkt ersichtlich.

Im Normal-Q-Q-Plot liegen die Punkte zwischen -2 und 2 annähernd auf der Gerade, aber außerhalb dieses Intervalls weichen sie stark ab von der Gerade. Das lässt darauf schließen, dass die Fehler nicht normalverteilt sind.

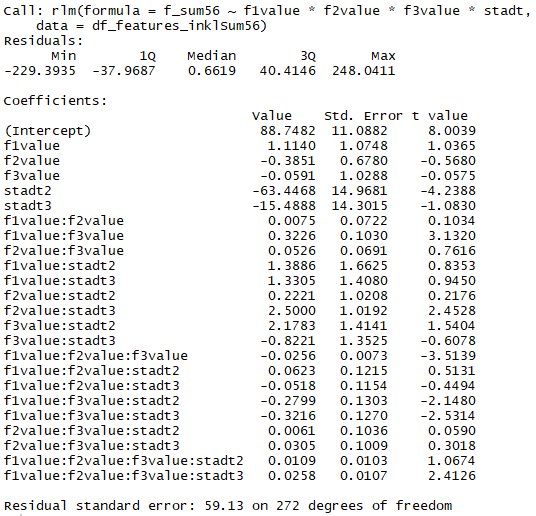
Der Scale-Location-Plot weist keine Muster auf.

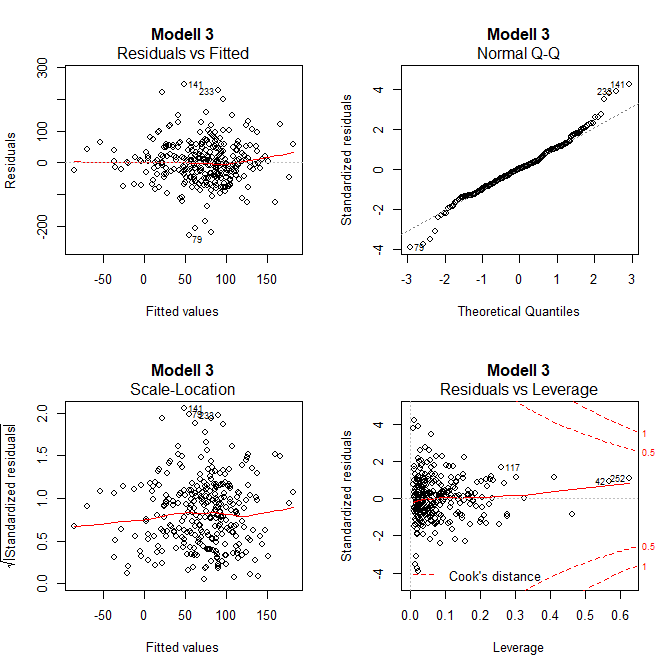
Der Residuals vs. Leverage Plot zeigt eine starke Dichte der Punkte am linken mittigen Rand. Die Punkte 42 und 252 liegen nahe an der Cooks Distance Linie. Daher sind sie als einflussreichere unter den Punkten des Plots zu beurteilen.

Lineares Modell – Versuch 3 (robustes Modell nach Huber-Ansatz):



Für das robuste lineare Modell wurde die Funktion „rlm“ verwendet, die als default-Belegung den Huber-Ansatz wählt. Der im Versuch 3 errechnete Residual standard error ist im Vergleich zu den beiden vorhergehenden Modellen geringer. Ein robustes Modell, wie z.B. nach dem Ansatz von Huber misst einzelnen Punkte nicht so große Bedeutung bei, sodass (einflussreiche) Outlier nicht so stark ins Gewicht fallen.





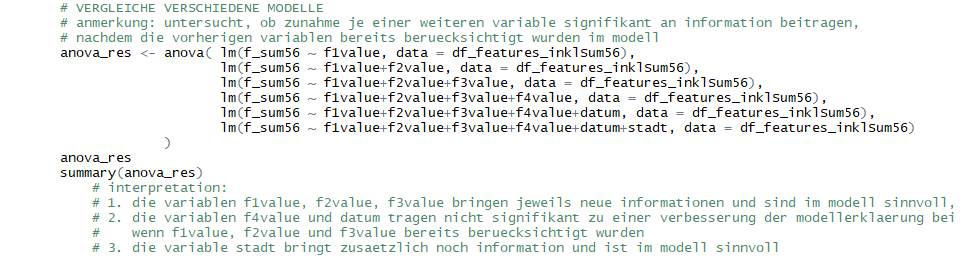
Hinsichtlich Residuals vs. Fitted Values ist im Vergleich zum ersten Modell eine stärkere Dichte der Punkte in der Mitte zu verzeichnen. Ein nichtlinearer Zusammenhang ist nicht direkt ersichtlich.

Im Normal-Q-Q-Plot liegen die Punkte zwischen -2 und 2 annähernd auf der Gerade, aber außerhalb dieses Intervalls weichen sie stark ab von der Gerade. Das lässt darauf schließen, dass die Fehler nicht normalverteilt sind.

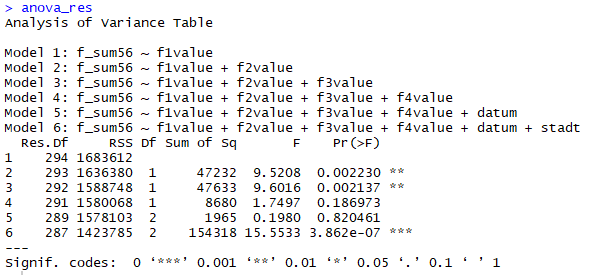
Der Scale-Location-Plot weist keine Muster auf.

Der Residuals vs. Leverage Plot zeigt eine starke Dichte der Punkte am linken mittigen Rand. Die Punkte 42 und 252 liegen nahe an der Cooks Distance Linie. Daher sind sie als einflussreichere unter den Punkten des Plots zu beurteilen.

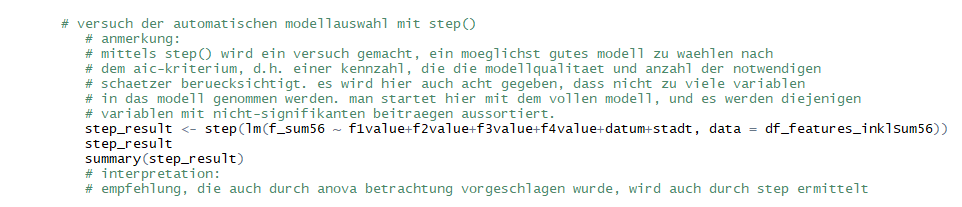
Vergleich verschiedener linearer Modelle:



Mittels der Funktion „anova“ wird die Varianzanalyse-Tabelle für 6 Modelle erstellt. Es wird untersucht, ob die Zunahme je einer weiteren Variable signifikant an Information beiträgt, nachdem die vorherigen Modellvariablen bereits berücksichtigt wurden. Der Ergebnis der ANOVA ist, dass die Variablen f1value, f2value und f3value jeweils neue Informationen bringen und im Modell sinnvoll sind. Die Variablen f4value und datum tragen nicht signifikant zu einer Verbesserung der Modellerklärung bei, wenn f1value, f2value und f3value bereits berücksichtigt wurden. Die Variable stadt bring zusätzlich noch Informationen und ist im Modell sinnvoll.



Versuch der automatischen Modellauswahl mit step():



Mittels der step-Funktion wird ein Versuch gemacht, ein möglichst gutes Modell zu wählen nach dem AIC-Kriterium, d.h. einer Kennzahl, die die Modellqualität und Anzahl der notwendigen Schätzer berücksichtig. Es wird hier auch Acht gegeben, dass nicht zu viele Variablen in das Modell genommen werden. Man startet mit dem vollen Modell und es werden diejenigen Variablen mit nicht-signifikanten Beiträgen aussortiert. Im vorliegenden Beispiel (für das lineare Modell mit additiven Zusammenhängen) werden also f1value, f2value, f3value und stadt2 für das Modell genommen. Die anderen Variablen werden aussortiert.

