Projektbericht:   
Statistische   
Datenanalyse des Diskmags-Projekts

Inhalt

[1. Einleitung 4](#_Toc155866408)

[2. Übersicht über die Daten 5](#_Toc155866409)

[3. Statistische Analyse der Daten 7](#_Toc155866410)

[3.1 Aufbau des DataFrame 7](#_Toc155866411)

[3.2 Auffinden einmalig vorkommender Daten 8](#_Toc155866412)

[3.3 Ausgabe fehlerhafter Einträge 8](#_Toc155866413)

[3.4 Untersuchung des Datensatzes auf Vollständigkeit 9](#_Toc155866414)

[3.5 Basis-Analyse der Daten mithilfe statistischer Verfahren 9](#_Toc155866415)

[3.6 Intensive statistische Analyse gewählter Kategorien 12](#_Toc155866416)

[3.6.1 Sprachen 14](#_Toc155866417)

[3.6.2 Herkunftsland 16](#_Toc155866418)

[3.6.3 Systeme 18](#_Toc155866419)

[3.6.4 Source 20](#_Toc155866420)

[3.6.5 Dauer 23](#_Toc155866421)

[3.6.6 Datum 24](#_Toc155866422)

[4. Beziehungen der Daten 28](#_Toc155866423)

[5. Vorhersagen auf die Daten mithilfe maschinellen Lernens 31](#_Toc155866424)

[6. Vorher vs. nachher: Die Veränderung der Datenzusammenhänge nach dem Update (20.10.2023) 34](#_Toc155866425)

[6.1 Datensatz 34](#_Toc155866426)

[6.2 Vollständigkeit der Zeilen 35](#_Toc155866427)

[6.3 Häufigkeit des Auftretens bestimmter Datensätze 36](#_Toc155866428)

[6.4 Datenvergleich: Sprache 37](#_Toc155866429)

[6.5 Datenvergleich: Herkunftsland 38](#_Toc155866430)

[6.6 Datenvergleich: Systeme 40](#_Toc155866431)

[6.7 Datenvergleich: Quellen 41](#_Toc155866432)

[6.8 Korrelations-Matrizen 42](#_Toc155866433)

[7. Probleme und Lösungen 43](#_Toc155866434)

[7.1 Datenzugriff 43](#_Toc155866435)

[7.2 Das Datum im Pandas Dataframe 43](#_Toc155866436)

[7.3 Zeichnen von Lebensdauer-Abschnitten 43](#_Toc155866437)

[7.4 Umwandlung der Kategorien in Zahlenwerte für das maschinelle Lernen 44](#_Toc155866438)

[7.5 Zu lange Spalten-Namen in der CSV-Datei 45](#_Toc155866439)

[7.6 Korrelations-Score 45](#_Toc155866440)

[8. Literaturverzeichnis 46](#_Toc155866441)

# 1. Einleitung

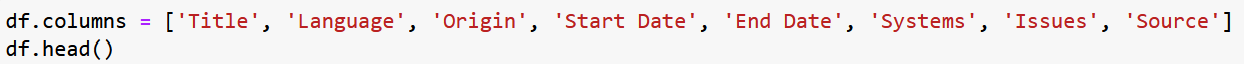
Einen kleinen USB-Stick schnell in den PC stecken und seine Daten abrufen. Mehrere Gigabyte abspeichern als Backup. Heutzutage kein Problem. Damals, in den frühen 1980er Jahren sah das noch ganz anders aus. Während auf einem dicken großen Röhrenbildschirm in geringer Hertz-Zahl die Zeichen flackernd und langsam auf dem Monitor erschienen, ein feiner kaum wahrnehmbarer Fiepton den stillen Raum erfüllte und das Atari, der Commodore 64 oder der MAC langsam starteten ohne den Luxus des bis zu 7000 MB/s schellen Lesezugriffs auf die NVME-SSD genießen zu können. Die aus unserer Sicht heute alte Technik, die wir nach 30 Jahren zu Recht als „Retro“ bezeichnen dürfen, war damals etwas so Besonderes und Zukunftsweisendes wie die Technik aus einem Star Trek Film. Doch soll es sich hier nicht um die gesamte Hardware drehen, sondern um die Disketten, ein ca. 13 x 13 cm großes schwarzes Rechteck mit einer Kapazität von 360 Kilobyte bis 1,2 Megabyte zum Speichern der Daten (später bis in die 2000er gab es dann die deutlich kleineren Floppy-Disks mit einer Kapazität von 720 KB bis sogar 2,88 MB).[[1]](#footnote-2) Als verbreitetes Speichermedium erreichte die Diskette in den späten 80er Jahren bis in die frühen 90er ihr Zenit an Popularität. Selbst heute gibt es vereinzelt immer noch begeisterte Sammler, die sich Retro-Computer oder deren leicht modernisierte Nachbauten kaufen, um Disketten einlegen und nutzen zu können. Heutzutage nicht mehr so häufig publiziert, ab der 80er aber stark präsent, waren die Diskmagazine, auch „diskmags“ genannt.[[2]](#footnote-3) Die Diskmags enthielten in der Regel immer eine Diskette mit Spielen, Informationen zum Magazin, das Magazin selbst, einfache Musik/Soundsamples oder kleine interaktive Anwendungen.[[3]](#footnote-4) Viele dieser Diskmags sind mittlerweile nicht mehr auf dem Markt. Aus diesem Grund wurde das Diskmags-Projekt der Universität Würzburg ins Leben gerufen, um das digitale Erbe der Zeitschriften am Leben zu erhalten (viele Bibliotheken lehnten eine Sammlung von Diskettenmagazinen ab[[4]](#footnote-5)). Die Daten zu den alten Zeitschriften kommen durch intensive Recherche bei Verlagen und unter Zuhilfenahme onlinebasierter Quellen zustande. Das Diskmags-Projekt ist verhältnismäßig jung, daher sind die Daten noch nicht vollständig, werden aber vom Projektteam regelmäßig aktualisiert. Im Zuge einer Projektarbeit im Hauptfach Digital Humanities sollen die vorhandenen Daten nun unter Verwendung statistischer Auswertungsverfahren genauer analysiert werden. Zur Auswertung der Daten wurde die Programmiersprache Python verwendet.

# 2. Übersicht über die Daten

Da die Daten in Form der csv-Datei eher unübersichtlich vorlagen, wurden die Informationen mithilfe von Python als Pandas Dataframe ausgelesen, um so eine gezielte tabellarische Übersicht zu erhalten. In der Regel reicht zum Auslesen eine grobe Übersicht über die ersten fünf Zeilen, daher bot sich die Verwendung der Funktion *.head()* an:  


Die folgende Auflistung zeigt die ersten fünf Zeilen. Hier kann man bereits erkennen, dass viele Bereiche noch als *NaN* markiert sind, d.h. es liegen zu diesem Datenpunkt noch keine genaueren Informationen vor, die dann erst durch ein zukünftiges Update ergänzt werden könnten. Ein Bild, das Text, Schrift, Zahl, Reihe enthält.

Automatisch generierte Beschreibung

Was gleich auffiel, waren die umständlichen Spaltenbezeichnungen *Magazine[„name“]*. Da es sich um sowieso um Magazine handelte und der zu lange Spaltentitel mit den eckigen Klammern eher für Irritationen beim Betrachter sorgt, wurden die Spalten unter Auslassung von *Magazine* einfach neu benannt:   
Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

In der vierten und fünften Spalte befinden sich *Start Date* und *End Date*. Da zur späteren statistischen Auswertung auch die genaue Lebensdauer der Magazine in Tagen interessant gewesen wäre, erhielt die Tabelle über Python-Code eine individuell hinzugefügte Spalte *Duration*. Hierzu musste das Datum erst einmal verarbeitbar gemacht werden, d.h. die Datumsangaben auch für Python als Datum erkenntlich gemacht werden. Über *to\_datetime()* und der Verwendung eines *formatters* war dies möglich. Nach der Formatierung erkannte Python die Datumsangaben als berechenbare Werte an, es konnte so die Differenz aus *End Date* und *Start Date* gebildet werden. Als Ergebnis entstand die neue Spalte *Duration*, die automatisch *days* ins Ergebnis mitaufnahm.



Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Nachdem nun formal eine statistisch verarbeitete Tabelle entstanden ist, ergibt es sicherlich Sinn, die Daten auch im Einzelnen zu betrachten. Über User-Input lassen sich beispielsweise spezifische Datenbereiche genauer observieren. Einzugeben sind entsprechend der Spaltenbezeichnung *Title, Language, Origin, Systems* oder *Source*. Das Ergebnis ist eine Tabelle, die nur die entsprechende Kategorie zur näheren Betrachtung ausgibt.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Bisher zeigten sich die Daten über Code-Ausgabe nur in statischer Form. Python bietet jedoch über eine Programmbibliothek, die sogenannten *ipywidgets* die Möglichkeit einer interaktiven Datenabfrage im Schnellzugriff mit *@interact*. Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

Der Codebetrachter braucht somit einfach über die Python-Oberfläche nach der Code-Ausführung im Feld *days* eine numerische Eingabe inklusive des Strings *days* zu machen, um den entsprechenden Startwert in Tagen eingeben zu können. Im Beispiel erscheint nach Eintippen von *11000 days* jeder Eintrag, der Magazine enthält, die länger als 11000 Tage bestanden.Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

# 3. Statistische Analyse der Daten

Bislang behandelte dieser Projektbericht nur die bloße Darstellung der Daten in Tabellenform. Unter Punkt 3 soll es hier um eine eingehendere statistische Betrachtung und erste Auswertungen gehen. Wir befinden uns nach getaner Vorarbeit, dem sogenannten *preprocessing* (Säuberung und Organisation der Daten), also schon im Hauptteil dieses Berichts, wo sich nun alles um die Datenanalyse dreht. Eng damit verknüpft ist auch Punkt 4, der die Beziehungen der Daten präziser betrachtet, jedoch zur Hervorhebung einzelner Aspekte ganz bewusst nicht in Punkt 3 integriert wurde.

## 3.1 Aufbau des DataFrame

Bevor wir tiefer in die Daten eintauchen, ist es grundsätzlich einmal interessant, wie der ganze Datenrahmen überhaupt aufgebaut ist. Die Funktion *df.shape* bietet sich dafür an, um überhaupt erst einmal zu sehen, wie viele Zeilen und Spalten der Datenbereich abdeckt. Die *shape*-Funktion offenbarte insgesamt 2501 Zeilen und 9 Spalten. Unser Datenrahmen besteht also aus 2501 verschiedenen Magazin-Titeln die über 9 Kategorien näher beschrieben werden.

## 3.2 Auffinden einmalig vorkommender Daten

Als statistische Voranalyse waren in den jeweiligen Spalten – sinnvoll ist hier die Wahl der Spalten *Language, Origin, Systems* und *Source* - zunächst die einmalig vorkommenden Daten kategorisch über die *.unique()*-Funktion gesondert auszugeben. Leere Einträge erschienen im Code-Ergebnis als *nan*. Hingen zwei oder mehrere Sprachen, Quellen oder Systeme zusammen, trennte lediglich ein Semikolon die Einträge. Wegen ihrer besseren Übersichtlichkeit sind wie folgt nur *Language* und *Origin* aufgeführt:

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Wie der Screenshot des Codes darüber zeigt, sind manche Magazine nicht nur monolingual, sondern auch bilingual erschienen (z.B. Englisch/Russisch, Englisch/Slowenisch, Englisch/Französisch etc).

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

## 3.3 Ausgabe fehlerhafter Einträge

Bei der Untersuchung der einmalig vorkommenden Länder zeigt sich ein weiterer Vorteil der *unique()*-Funktion: Es lassen sich auch Tipp- bzw. Kopierfehler des Erstellers der Daten erkennen. Wenn man die Ausgabe der *.unique()* von *Origin* genauer betrachtet, fällt auf, dass sich versehentlich eine Datumsangabe (*08.09.1997*) sowie die Sprachbezeichnungen *German* und *Polish* miteingeschlichen haben. Das Problem solcher Fehler ist, dass sie am Ende die Auswertung der gesamten statistischen Werte beeinflussen. Bei vielen Werten ist die Einflussnahme fehlerhafter Werte zwar vernachlässigbar, aber trotzdem störend. Aus diesem Grund ist auch an dieser Stelle eine Säuberung der Daten (*data cleaning*) erforderlich. Dass und vor allem welche Art von Falscheinträgen vorhanden sind, ist bekannt, allerdings nicht an welcher Stelle bzw. an welchem Index der Fehler auftritt. Zur exakten Indexausgabe bietet ein Pandas-Dataframe die Möglichkeit eine *.query()*-Funktion zu verwenden. Nach Anwendung von *query()* werden nun die gesuchten Einträge entsprechend ausgegeben:  
Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Reihe, Schrift, Screenshot enthält.

Automatisch generierte Beschreibung

Diese Falsch-Einträge werden später bei der statistischen Auswertung im Code berücksichtigt und entsprechend eliminiert.

## 3.4 Untersuchung des Datensatzes auf Vollständigkeit

Zum Anzeigen aller Daten, die keine *NaN*-Einträge besitzen (*NaN* steht für *not a number* und verweist auf einen leeren Eintrag innerhalb der Tabelle) wurde die *dropna()*-Funktion angewendet. Der Code-Auszug unten, realisiert durch einen Screenshot, zeigt, dass von 2501 Datenzeilen, zumindest 366 Zeilen (= 366 Datensätze) keine Leereinträge aufweisen, was einer Vollständigkeit von gerade einmal 14,64 % des gesamten Datensatzes entspricht. Das mag zunächst sehr wenig erscheinen, allerdings ist zu bedenken, dass das Diskmags-Projekt noch verhältnismäßig jung ist und noch viele Update-Phasen vor sich hat.

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

## 3.5 Basis-Analyse der Daten mithilfe statistischer Verfahren

Mithilfe der Python-Funktion *.describe()* lassen sich zunächst einige grundlegende Statistiken zu den Kategorien darstellen. Dazu gehören die Anzahl, wie viele einzigartig Daten enthalten sind sowie die am häufigsten vorkommenden Bestandteile von *Language, Origin, Systems* und *Source*:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Wie die Code-Ausgabe oben zeigt, enthält der Datensatz 2501 Magazin-Titel insgesamt, die alle auch einmalig vorkommen. Wie unter 3.1 erwähnt, besteht der Datenrahmen aus insgesamt 2501 Zeilen, d.h. wie zu erwarten befinden sich bei *Title* keine *NaN*-Einträge. Die Angabe *top* ist hier nur zufällig vom System ausgewählt und hat die Funktion eines Platzhalters, da alle Werte je einmal vorkommen bei den Titeln. Interessanter ist es dann schon eher in den anderen Kategorien. *Language* zählt 1161 Sprachen (der Rest ist wie schon erwähnt ein noch nicht bekannter Leer-Eintrag) mit 25 *unique*-Werten. Es befinden sich also 25 verschiedene Sprachen bzw. Kombinationen von zwei Sprachen im Datensatz. Die am häufigsten vorkommende Sprache ist Englisch, 654-mal vorkommend. Das Herkunftsland *Origin* verfügt über 448 bekannte Einträge mit 33 verschiedenen angegebenen Ländern, darunter 214-mal Russland als top Wert. Unter *Systems* (2492 Werte) gibt es wenige Fehleinträge (neun) mit 64 unikalen Datensätzen, wobei hier noch zu erwähnen ist, dass mehrere Systeme möglich waren und dadurch unterschiedliche durch Komata getrennte Kombinationen entstanden, die einen solch hohen Wert erklären. In Wirklichkeit sind es vermutlich nicht mehr als zehn Systeme in der *unique*-Zählung. Das Commodore 64 wies dabei die höchste Zahl auf. Die Quelle *Source* ist wiederum vollständig mit 2501 Werten. Wie bei den Systemen, sind auch bei den Quellen mehrere Quellenangaben nebeneinander möglich, was die hohe Anzahl von 29 Quellen erklärt. Einzeln kommt *Demozoo* am häufigsten vor mit 729-mal.

Der Absatz zuvor erklärt bislang nur fünf Kategorien näher und berücksichtigt ganz bewusst noch nicht *Start Date, End Date* und *Duration*, da hier Anzahl (*count*), Einzigartigkeit (*unique*), Top-Werte (*top)* und Häufigkeit (*freq*) wenig Sinn ergeben würden. Die *describe*-Funktion() gibt bei den Datumsangaben zudem andere statistische Auswertungen zurück als bei den zuvor genannten Kategorien, aus diesem Grund wurde *describe* jeweils auf jede entsprechende Kategorie einzeln angewendet. Die Rückgabe des Codes behält zwar noch die Gesamtzahl *count*, jedoch fallen *unique, top* und *freq* weg. Neu hinzu kommen jetzt der Mittelwert *mean*,die Quartile, mit je 25, 50 und 75 Prozent sowie Minimum und Maximum. Die Quartile bezeichnen

*die Differenz zwischen dem oberen und dem unteren Quartil in einer Menge sortierter Werte. Quartile werden gebildet, indem eine Menge sortierter Werte in vier gleich große Gruppen unterteilt wird. Das obere und das untere Quartil begrenzen die mittleren 50 % der Werte innerhalb einer Variablen. Das bedeutet, 50 % aller Variablenwerte liegen zwischen dem oberen und dem unteren Quartil. […] Der erste Schritt bei der Bestimmung des oberen und unteren Quartils (manchmal auch 75-%und 25-%-Perzentil genannt) besteht darin, Ihre Werte vom niedrigsten zum höchsten zu sortieren. Anschließend bestimmen Sie diejenige Position in der Wertemenge, unter der 25 % (oder ein Viertel) der Werte liegen. Der Wert an dieser Position ist das untere Quartil* *(25-%-Perzentil). Suchen Sie nun die Position in der Wertemenge, über der 25 % (oder ein Viertel) der Werte liegen. Der Wert an dieser Position ist das obere Quartil (75-%-Perzentil). Die Positionen des unteren und oberen Quartils in einer sortierten Wertemenge können Sie ganz einfach mithilfe der folgenden Formel finden: Position des unteren Quartils = ( n + 1)÷4, wobei n die Anzahl der Werte in der Datenmenge ist. Position des oberen Quartils = 3( n + 1)÷4, wobei n die Anzahl der Werte in der Datenmenge ist. Bei 20 Werten lautet die Formel also: Position des unteren Quartils = (20 + 1)÷4 = 21÷4 = 5,25 Position des oberen Quartils = 3(20 + 1)÷4 = 63÷4 = 15,75 Diese Ergebnisse sind die Positionen, an denen das obere und das untere Quartil zu finden sind, nicht jedoch die eigentlichen Werte des oberen und unteren Quartils![[5]](#footnote-6)*

Laut *Start Date* kam das am frühesten publizierte Magazin im Januar 1981 auf den Markt, das späteste startete sogar März 2023. Der Mittelwert beziffert sich auf das Jahr 1995:

Ein Bild, das Text, Schrift, Screenshot, weiß enthält.

Automatisch generierte Beschreibung

Beim *End Date* ist ebenfalls Januar 1981 verbucht (entspricht auch dem Wert in *Start Date*), allerdings ist hier anzumerken, dass dies vermutlich eher am noch unvollständigen Datensatz liegt und hier noch keine genauere Information gefunden werden konnte. Das zuletzt erschienene Magazin liegt im April 2023. Der Durchschnitt ist etwas höher als beim *Start Date* und weist auf 1996 hin:

Ein Bild, das Text, Schrift, Screenshot, weiß enthält.

Automatisch generierte Beschreibung

Die Dauer *Duration* zeigt einen Mittelwert von 458 Tagen an. Am längsten hielt sich ein Magazin für 11715 Tage auf dem Markt. Dennoch gibt es zu viele 0-Tage-Einträge, was sich am ersten Quartil, dem 25-Prozent-Quartil offenbart, wo ebenfalls 0 Tage aufgrund fehlender Vollständigkeit der Daten verbucht wurden:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

## 3.6 Intensive statistische Analyse gewählter Kategorien

Im Folgenden geht es um eine genauere Untersuchung der Spalten *Languages, Origin, Systems, Source* und *Duration*. Der nach diesem Absatz eingefügte Code-Abschnitt behandelt hier zwar nur die *Languages*, wird jedoch in gleicher Weise auf die anderen Kategorien angewendet (statt *Languages* steht dann stattdessen einfach z.B. *Source* imf DataFrame *df*), bis auf *Duration*, da hier noch ein wenig zusätzlicher Anpassungsaufwand nötig war. Der Code ist so aufgebaut, dass zunächst die Werte der jeweiligen Kategorie absteigend geordnet werden, beginnend mit dem am meisten vorkommenden Wert. Am Ende des Codes wurden mit Hilfe von Diagrammen über *matplotlib* die Zahlenwerte sowie die prozentualen Werte im Stil einer Heatmap (Rot hat die höchsten Werte, Dunkelblau die niedrigsten) visualisiert. Danach kommt es zu einer Auswertung der fehlenden Werte (*NaN entries*), gefolgt von einer Auflistung der statistischen Größen *Mittelwert*, *Median* und *Standardabweichung*. Während der Mittelwert den einfachen Durchschnitt repräsentiert, ist der Median am Beispiel „Geschlechtsrollenorientierungen […] – er liegt beim ersten Item bei der Ausprägung „stimme eher nicht zu“ und bei den beiden anderen bei „stimme überhaupt nicht zu“. Wenn man an dieser Stelle stehen bleibt, missachtet man aber, dass die Antwortmöglichkeiten ja eine gewisse Rangfolge beinhalten, hier eben eine ansteigende Zustimmung zu einem eher traditionellen Familienbild. Um diese Eigenschaft der Messung zu berücksichtigen, verwendet man den Median einer Verteilung. Der Median ist der Wert, der eine geordnete Verteilung in zwei gleichgroße Hälften unterteilt. Bei allen drei Items ist dies der Wert 2“.[[6]](#footnote-7) Die Standardabweichung gilt als die „Größe zur Quantifizierung der Streuung“[[7]](#footnote-8) der Daten von ihrem Mittelwert oder wie Hartmann/Kopp/Lois etwas schöner definieren: „Die Streuung der erwähnten Normalverteilung wird auch als *Standardfehler der Differenz* bezeichnet. Allgemein gesprochen ist der Standardfehler die Standardabweichung einer Stichprobenverteilung, d.h. der Verteilung eines Stichprobenkennwerts. […] Der Standardfehler gibt damit an, wie stark Stichprobenkennwerte um den wahren Wert in der Population streuen.“[[8]](#footnote-9)

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

### 3.6.1 Sprachen

Nach Ausführung der *value\_counts()*-Funktion zeigt sich, dass wenig überraschend Englisch die meist verwendete Sprache war (654-mal), gefolgt von Russisch (300) und Polnisch immerhin an vierter Stelle (46), knapp gefolgt von Deutsch (42):

Ein Bild, das Text, Screenshot, Schrift, Karte Menü enthält.

Automatisch generierte Beschreibung  
Bei den Leereinträgen ergaben sich 1339 fehlende Werte, das macht immerhin einen Anteil von 53,56% des gesamten aktuell verfügbaren Datensatzes aus.

Der Mittelwert liegt bei 46,44, der Median bei 2,0 und die Standardabweichung ca. bei 140. Vor allem bei der Standardabweichung merkt man hier bei den Werten der *Languages* die hohe Streuung um den Mittelwert.

Visualisiert stellen sich die Sprachen nach Werten und Prozentzahlen so dar:

Ein Bild, das Text, Screenshot, Diagramm, parallel enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

### 3.6.2 Herkunftsland

Das Haupt-Herkunftsland ist erstaunlicherweise Russland (214) gefolgt von Belarus (52) und Polen (51), dahinter wieder Deutschland (40). Aus den USA kommen nur 4 Zeitschriften. Dass die USA hier so gering vertreten ist, liegt sicherlich nicht daran, dass es dort nicht viele Zeitschriften gab, sondern vermutlich eher an den noch nicht vollständigen Daten. Nach ein paar Monaten von Aktualisierungen für das Diskmags-Projekt kann es in Zukunft hier sicherlich zu einer deutlicheren Verschiebung kommen. Nach dem letzten Oktober-Update für Diskmags ließ sich ebenfalls eine statistische Änderung der Datenlage erkennen (siehe Abschnitt 6).

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Die Herkunftsländer sind der nach aktuellem Stand wohl größte Bereich, der noch Aktualisierungsbedarf hat. Insgesamt fehlen 2053 Werte, das macht einen Prozentsatz von 82,12 % des gesamten Datensatzes aus.

Der Mittelwert liegt bei 14,9, der Median ist wie bei den Sprachen wieder bei 2,0 und die Standardabweichung fällt jedoch verhältnismäßig gering aus bei den Herkunftsländern mit ca. 40.

Grafisch dargestellt erscheinen die Herkunftsländer wie folgt (Zahlen gefolgt von Prozent):

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

### 3.6.3 Systeme

Da die Auflistung andernfalls zu lange geworden wäre, wurden zur besseren Übersicht nur alle Werte größer als 1 berücksichtigt. In den Berechnungen sind wiederum alle Werte, auch kleiner als 2, nach wie vor enthalten. Wie in der nachfolgenden Auflistung zu erkennen ist, waren unter den Zeitschriften mit den Retro-Systemen das Commodore 64 am weitesten verbreitet mit 783 Systemen. An zweiter Stelle rangieren MS-DOS-Systeme (540), gefolgt von AMIGA OCS/ECS (349) und ZX Spectrum (327), etwas dahinter das ATARI ST/E. Interessanterweise schaffte es sogar Windows noch in die Top 10 mit 32 Vertretern. Manche der Disketten bzw. CDs waren auch auf mehreren Systemen lauffähig, weswegen auch in der Kategorie *Systems* wieder Mehrfachkombinationen (Mehrfachnennungen) zustande kamen.

Ein Bild, das Text, Screenshot, Karte Menü, Dokument enthält.

Automatisch generierte Beschreibung

Die Vollständigkeit der Daten in der System-Spalte ist durchaus gegeben, da lediglich 9 fehlende Daten gefunden werden konnten, was gerade einmal 0,36 % fehlender Values entspricht.

Der Durchschnitt der Systeme beziffert sich recht niedrig mit 38,92 Systemen, der Median liegt bei 2,0 und die Standardabweichung fällt aufgrund dessen sehr hoch aus mit 129,61.

Die grafische Darstellung gestaltete sich als nicht ganz einfach wegen der vielen Systeme und deren zahlreichen Kombinationen, über mehrere Anpassungen konnte jedoch eine noch verhältnismäßig gute Visualisierung realisiert werden:

Ein Bild, das Text, Screenshot, Diagramm, parallel enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Diagramm enthält.

Automatisch generierte Beschreibung

### 3.6.4 Source

Bei den Quellen waren in der Regel häufig Mehrfachnennungen möglich, dennoch zeigte sich deutlich *Demozoo* als häufigste Quelle (728-mal), selbst an zweiter Stelle trat nochmals *Demozoo* auf, dieses Mal jedoch in Kombination mit *Pouet* (347). An dritter Stelle erschien das russische Magazin ZXpress (296), dahinter Internet Archive sowie Kombi InternetArchive/CSDB (beide je 176).

Ein Bild, das Text, Screenshot, Dokument, Schrift enthält.

Automatisch generierte Beschreibung

Die Daten lagen komplett vollständig vor, es gab keine fehlenden Werte.

Der Mittelwert bezifferte sich auf ca. 86,21, die Quelle *Pouet* lag damit also mit 86 nur knapp unter dem Durchschnitt. Der Median ergab 15,0 und die Standardabweichung fiel wieder verhältnismäßig hoch aus mit rund 154,71.

Grafisch sieht man nochmal besser, wie sich *Demozoo* unter den Quellen deutlich abhebt. Unter den Prozentwerten wirkt alles durch die hohe Anzahl der Quellen ziemlich gleichmäßig, wobei hier auch *Demozoo* heraussticht (s. nächste Seite).

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

### 3.6.5 Dauer

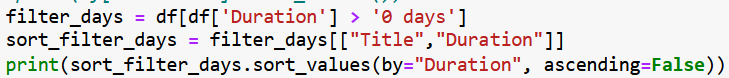
Die Spalte *Duration* war vorher nicht im Datensatz enthalten, sie wurde individuell erstellt und über Python aus den beiden Spalten mit den Datumsangaben *Start-Datum* und *End-Datum* für eine eingehendere genauere Untersuchung der Lebensdauer der Magazine berechnet. Zur besseren Übersichtlichkeit sind nicht alle Werte aufgeführt, sondern nur die Lebensdauer von mehr als 20 Jahren. An der Spitze befindet sich *Propaganda* mit einer Lebensdauer von 11715 Tagen, knapp dahinter gefolgt von *Vandalism News* und *Addybook*.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Es fehlen 284 Werte, das sind 11,36 Prozent.

Der Mittelwert liegt bei 1,85, der Median bei 1,0 und die Standardabweichung bei nur 2,33, was dadurch zu erklären ist, dass die Datumsangaben noch nicht richtig aktualisiert sind und wohl erst über zukünftige Updates aussagekräftiger werden. Die Lebensdauer von kleiner als 10 Tagen, nach Änderung des Codeausschnitts von *> 0 days* zu *< 10 days*, zeigt hier bereits 996 vorhandene Werte an, was logisch gesehen eigentlich nicht so viel Sinn ergibt.



### 3.6.6 Datum

Bei der Analyse des Datums waren einige gesonderte Anpassungen nötig, weshalb in diesem Unterabschnitt wieder eine code-nahe Erklärung zusammenhängend mit der statistischen Daten-Auswertung folgen soll.

Zunächst gab es für den Nutzer wieder die Möglichkeit über die Python-Oberfläche ein gewünschtes Datum einzugeben, um so eine Auflistung mit dem Startdatum zu realisieren. Nach dem Schema *YYYY-MM-DD*, z.B. die Datumseingabe *2020-03-24* für den 24.03.2020, ließ so alle Starteinträge filtern:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Ebenso wie bei den Kategorien zuvor, sollte auch beim Datum eine Untersuchung auf *NaN*-Einträge erfolgen. Das Ergebnis waren 281 Leereinträge beim *Start* *Date* und 284 Leereinträge beim *End Date*, was grob gerundet immer noch 10 Prozent der Gesamtdaten ausmacht (kein schlechter Wert aber immer noch zu viele *NaN*-Werte):

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

Bislang handelte dieser Abschnitt nur von der Auflistung der Daten. Jetzt soll es gezielt um die grafische Realisierung über *matplotlib* gehen. Das Problem beim Zeichnen war jedoch der große Umfang von knapp 2500 Daten. So viele Daten zu zeichnen, würde schnell unübersichtlich werden. Aus diesem Grund sollte wieder ein User-Input vorgenommen werden, sowohl für die Eingabe des Herkunftslandes als auch der Lebensdauer in Tagen. Nach der User-Eingabe *Germany* ließen sich aufgrund des kleinen Umfangs deutscher Magazine, recht übersichtlich die verschiedenen Titel ausgeben. Am längsten hielt sich das Magazin *Magic Disk* mit 3287 Tagen gefolgt von *Golden Disk* (2922) und *PC Action* (1826):

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Nach der Untersuchung der Lebensdauer nach Tagen, sollte auch die Dauer in Jahren untersucht werden. Dazu musste aus den jeweiligen Datumsspalten *Start Date* und *End Date* die Jahre entsprechend über die Funktion *.dt.year* gefiltert werden. Um die Jahresdauer zu integrieren (Bildung der Differenz aus *Start Year* und *End Year*) wurde dem Datenrahmen eine extra Spalte *Duration (years)* hinzugefügt. Das Ergebnis ist auf der Folgeseite nach dem Code dargestellt.

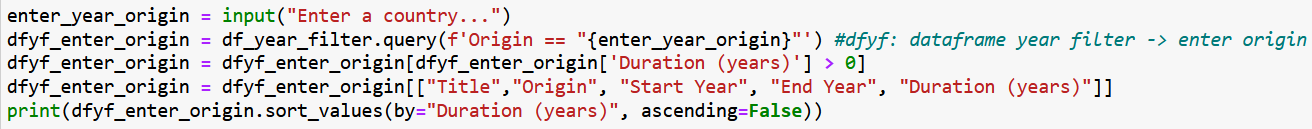
Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Wie an der Code-Ausgabe oben zu sehen ist, sind 2216 Zeilen nur schwer darstellbar, vor allem wenn es um die grafische Umsetzung im Diagramm geht. Daher bot sich auch hier wieder ein User-Input an. Als Eingabe wurde für dieses Beispiel *Belarus* gewählt.



Ein Bild, das Text, Screenshot, Karte Menü, Zahl enthält.

Automatisch generierte Beschreibung

Wie die Ausgabe oben zeigt, konnte sich *C-Net Week* ganze 10 Jahre halten, mit etwas Abstand *Target* (7), *Body* (6), dahinter mit jeweils 4 Jahren *Echo*, *Impulse* und *RIP*. Eine nachfolgende Auswertung der Basis-Statistiken über die *.describe()*-Funktion ergab eine Anzahl von 18 Zeitschriften aus Belarus. Das durchschnittliche Startjahr war 1997, das Endjahr 2000. Es gab nur minimale Standardabweichungen (0,922 Start und 2,66 End). Die früheste Aufzeichnung eines belarussischen Magazins datiert auf das Start-Jahr 1996, das End-Jahr 1997. Spätestes Startjahr war 1999, das End-Jahr geht auf 2007 zurück.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Um das ganze etwas übersichtlicher zu gestalten wurden mit *matplotlib* die Diagramme Publikationsdauer (links oben), Startjahre (rechts oben), Endjahre (links unten) sowie eine Gegenüberstellung von Startjahren und Endjahren (rechts unten) der Magazine:

|  |  |
| --- | --- |
| Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.  Automatisch generierte Beschreibung | Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.  Automatisch generierte Beschreibung |
|  |  |

# 4. Beziehungen der Daten

In diesem Abschnitt sollen die Relationen der Daten zueinander untersucht werden. Zunächst erscheint eine Auflistung der kategorialen Kombinationen *Language – Origin*, *Origin – Systems*, *Origin – Source* und *Language – Source*. Anschließend soll eine Korrelationsmatrix die Beziehungen näher beleuchten.

Ein Bild, das Text, Karte Menü, Screenshot, Dokument enthält.

Automatisch generierte Beschreibung

Die Code-Ausgabe (Bild links) listet eine Aufzählung von Sprache zusammen mit Herkunftsland auf. Wie man sehen kann, dominiert klar Russisch (Russland und Belarus) mit 210+52 Zeitschriften, danach Polnisch/Polen mit 38, dann schon Deutsch/Deutschland mit 30 Werten. Englisch ist aufgrund der Datenlage noch verhältnismäßig selten vertreten, auch in Kombination mit anderen Sprachen. Zukünftige Daten-Updates könnten hier nochmal für eine Verschiebung sorgen.

Ein Bild, das Text, Screenshot, Karte Menü, Schrift enthält.

Automatisch generierte Beschreibung

Bei *Sprache-System* ist vor allem im osteuropäischen Raum am häufigsten *ZX Spectrum* vertreten. Eine Ausnahme bildet Großbritannien, denn auch hier wurde *ZX Spectrum* verwendet. Ansonsten traten sehr häufig *Commodore 64, Amiga* sowie *MS-DOS* auf. Da es sehr viele 1er-Einträge gab, wurden nur alle Ergebnisse größer als 1 ausgegeben.

Ein Bild, das Text, Karte Menü, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Auch bei den Quellen in Relation zum Herkunftsland ist ablesbar, dass Osteuropa von bestimmten eigenen Quellen dominiert wurde. Hier zeigte sich vor allem *ZXpress* als Hauptquelle. Westliche Quellen beruhten hauptsächlich auf *Demozoo, Pouet* oder *Internet Archive*.

Ein Bild, das Text, Dokument, Karte Menü, Screenshot enthält.

Automatisch generierte Beschreibung

Bei *Language-Source* dominiert klar *ZXpress* mit Russisch (280). Russisch kam jedoch auch in ein paar wenigen anderen Quellen(kombinationen) vor, wie z.B. *Pouet/ZXpress* oder *Demozoo / Pouet / ZXpress.* Bei Deutsch war es vor allem *Kultboy*, bei Englisch in der Regel immer *CSDB* und *Internet Archive*. Auffallend ist hier zudem die Anzahl der Quellen englischer Sprache, die hier zumindest in der Summe die russischen übertrifft (148+143+114+104+42+23+…).

Eine Korrelationsmatrix stellt im Folgenden nochmals genauer die Korrelationen dar, jedoch ist hier zu beachten, dass bei dieser Ergebnismatrix die *NaN*-Einträge entsprechend gefiltert wurden. Die Matrix ist eigentlich Teil von Abschnitt 5 *Vorhersagen auf die Daten mithilfe maschinellen Lernens*, weswegen in diesem Paragraph noch nicht genauer auf Code und Umsetzung eingegangen wird. Eine Heatmap (s. nächste Seite) visualisiert die besten Korrelationen von Blau (Korrelation nahezu Kalt) über Rot (mittelmäßig), bis hinzu Gelb (sehr hoher Korrelationswert). Die 1.0-Werte sind hier allerdings identische Kategorien (z.B. *Title-*Title) und können ignoriert werden. Größte Übereinstimmungen repräsentieren die kategorialen Kombinationen von *Language-Source* (hellorange), gefolgt von *Language-Systems* und *Source-Systems* (je dunkelorange).

Ein Bild, das Text, Screenshot, Zahl, Farbigkeit enthält.

Automatisch generierte Beschreibung

# 5. Vorhersagen auf die Daten mithilfe maschinellen Lernens

Im Folgenden soll untersucht werden, inwieweit es Sinn gemacht hätte, Vorhersagen über maschinelles Lernen aufgrund der Datenbasis zu treffen. Zunächst war es nötig, die *NaN*-Einträge über *.dropna()* herauszufiltern. Das Ergebnis waren 366 vollständige Zeilen:

Ein Bild, das Schrift, Text, Reihe, Grafiken enthält.

Automatisch generierte Beschreibung

Des Weiteren wurden den Daten innerhalb der Kategorien Nummern zugewiesen, um für spätere Vorhersagen die Zuordnungen gleich erkennen zu können (so repräsentiert die Zahl *9* in *Source* die Quelle *Pouet*). Die zuvor zugewiesenen Nummern innerhalb der Daten je Kategorie lagen jedoch noch normal als String, also als Name der jeweiligen Zeitschrift, Land, Quelle etc. vor, die Kategorien mussten somit über die Funktion *pd.factorize* in Zahlen konvertiert werden:

Ein Bild, das Text, Schrift, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Wie am Screenshot direkt über diesem Absatz zu erkennen ist, repräsentiert jetzt jede Zahl, beginnend von 0 bis zur letzten Unterschiedlichen Datenbezeichnung einen Eintrag in der Tabelle. Gleiche Zahlen sind dementsprechend identische Einträge.

An dieser Stelle erfolgte auch die Betrachtung der Korrelationsmatrix (s. Abschnitt 4), die die Beziehungen der Kategorien zueinander prozentual darstellte. Über die Funktion *.corr()* sowie die Python-Bibliothek *seaborn* ließ sich mit wenig Aufwand, angewendet auf die von *NaN-*Einträgen gereinigte Datentabelle, eine Korrelatonsmatrix berechnen und zeichnen.

Ein Bild, das Text, Schrift, Screenshot, weiß enthält.

Automatisch generierte Beschreibung

Die eigentliche Voraussage auf die Daten erfolgte über die Anwendung einer logistischen Regression, da dieses Model in der Regel sehr schnell und präzise Daten mithilfe maschinellen Lernens verarbeitet. Die logistische Regression definiert Jung[[9]](#footnote-10) wie folgt: „Logistic regression is a method for classifying datapoints which are characterized by feature vectors x ∈ Rn (feature space X = Rn) according to two categories which are encoded by a label y. It will be convenient to use the label space Y = R and encode the two label values as y = 1 and y = -1. Logistic regression learns a hypothesis out of the hypothesis space H(n)“. Im Code werden die Daten auf Basis der Kategorien *Title, Source, Origin* und *Systems* trainiert und als Vorhersage zurückgegeben. Das Resultat ist eine Liste sortiert nach der Reihenfolge des Einlesens der Daten und der jetzt erfolgten Vorhersage in numerischer Form. Zunächst der Code, im Anschluss die Liste als Ergebnis, gezeigt am Beispiel von *Origin*. Die vorherzusagenden Kategorien konnten per User-Input eingegeben werden (*hier: Origin*):

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Schrift, Typografie enthält.

Automatisch generierte Beschreibung

Nach Berechnung der vorhergesagten Werte, ließ sich darüber allerdings noch keine genaue Erkenntnis über die Zuverlässigkeit der Vorhersagen ziehen. Aus diesem Grund wurde mit Hilfe von *sklearn.metrics* ein *accuracy/precision/recall/f1* score berechnet. Die Genauigkeit zeigte zwar einen verhältnismäßig guten Wert mit ca. 78,47 Prozent, Präzision, Recall und F1 jedoch sehr schwache Werte, alle zwischen 25 und 30 Prozent. Die niedrigen Werte lassen sich so erklären, dass wegen des zu kleinen Datensatzes mit übermäßig vielen verschiedenen Werten (nur bei mehreren identischen Werten sind auch genauere Ergebnisse erzielbar) eine nicht allzu exakte Vorhersage getroffen werden konnte. Parametertunings über Parameter wie *micro*, *macro* und *weighted* (DefiniEin Bild, das Text, Schrift, Screenshot, Algebra enthält.

Automatisch generierte Beschreibungtion, s. Khalusova[[10]](#footnote-11)), nahmen zumindest minimal Einfluss auf das erzielte Ergebnis:

# 6. Vorher vs. nachher: Die Veränderung der Datenzusammenhänge nach dem Update (20.10.2023)

Im Folgenden sollen die Veränderungen nach dem Oktoberupdate vom 20. Oktober 2023 dokumentiert werden. Dabei geht es vor allem um die Fragestellung, inwieweit sich das Update auf Datensatz, Vollständigkeit der Daten sowie statistische Auffälligkeiten bzw. Wertveränderungen ausgewirkt hat. Aus diesem Grund erscheint ein Vorher-Nachher-Vergleich am sinnvollsten.

## 6.1 Datensatz

Zunächst vergrößerte sich der Datensatz um 24 neue Einträge, somit um ca. 1 Prozent (erkennbar an der Anzahl der zusätzlichen Zeilen). Das erscheint im ersten Moment nicht viel, bei regelmäßigen zukünftigen Updates könnte dies allerdings die Anzahl der Daten auf wertvolle Weise erweitern.

Datensatz vor dem Update (2477 Zeilen):

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

Datensatz nach dem Update (2501 Zeilen):

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

## 6.2 Vollständigkeit der Zeilen

Nach dem Update konnten insgesamt 366 vollständige Zeilen, d.h. Zeilen, die keine *NaN*-Einträge enthalten, erfasst werden. Im Vergleich zu vorher ein Sprung um 120 Zeilen, wobei hier zu berücksichtigen ist, dass sich der Datensatz vergrößert hatte. Prozentual gesehen enthielt der Datenrahmen vorher ca. 9,931 Prozent vollständiger Datenzeilen (246 : 2477), nachher immerhin schon 14,634 Prozent (366 : 2501). Die Aktualisierung lückenhafter Einträge befindet sich somit bereits auf einem guten Weg.

Vor dem Update:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Nach dem Update:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

## 6.3 Häufigkeit des Auftretens bestimmter Datensätze

Wie am Vergleich zwischen vorher und nachher zu sehen ist, wurden vor allem in den Spalten *Language* und *Origin* Einträge ergänzt, vor allem bei *Language* war eine Veränderung von 393 auf 1161 Nicht-*NaN*-Daten zu verzeichnen. Auch die Anzahl der Sprachen bzw. deren Kombinationen mit anderen Sprachen erhöhte sich um zehn (15 auf 25). Bei *Origin* erhöhte sich die Anzahl von Herkunftsländern bzw. weitere Herkunftsländer in Kombination sogar von 14 auf 33. Interessant ist vor allem die Werteveränderung. Nachdem vor dem Oktoberupdate noch Russisch mit 276-mal am häufigsten auftrat, ergab die Aktualisierung der Daten, dass jetzt Englisch mit 654 Einträgen den neuen top Wert einnahm. Das häufigste auftretende Herkunftsland ist jedoch immer noch Russland (vorher: 142, nachher: 214). Bei den Systemen dominiert nach wie vor das Commodore 64. Eine Veränderung ergab sich dann auch bei den Quellen, wo *ZXpress* mit 265-mal von *Demozoo* (729-mal) abgelöst wurde.

Vor dem Update:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

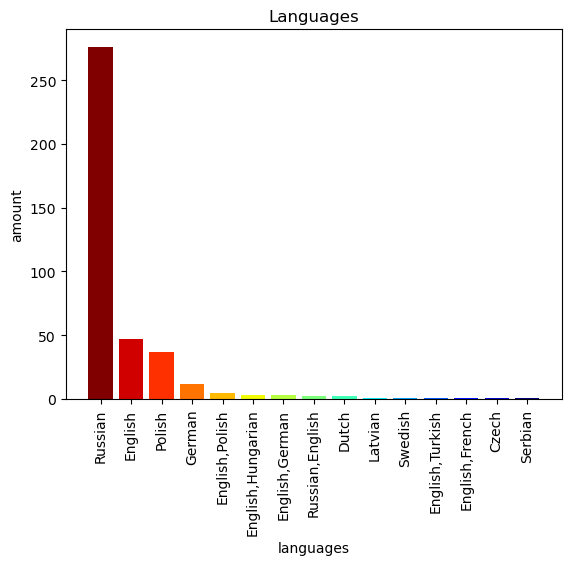
Nach dem Update:

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

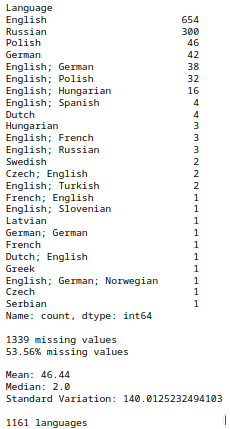
Automatisch generierte Beschreibung

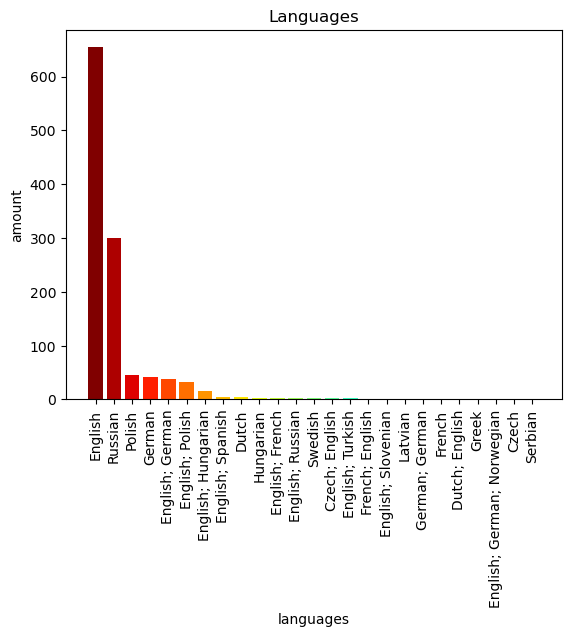
## 6.4 Datenvergleich: Sprache

Die nachfolgenden Screenshots betrachtend, ist zu sehen, dass eine doch deutliche Aktualisierung vor allem bei Englisch stattgefunden hat. Von zuvor 47 auf jetzt 654, liegt jetzt ein realistischer Wert vor, da Englisch als Weltsprache die größte Auftrittswahrscheinlichkeit haben sollte. Durch die ergänzten Daten nach dem Update haben sich zudem neue Kombinationen von Sprachen ergeben. So erscheint jetzt Englisch/Deutsch/Norwegisch oder Englisch/Slowenisch neu in der Liste. Des Weiteren haben sich viele Werte deutlich erhöht, wie Deutsch oder Englisch/Polnisch. Auch statistisch gesehen kam es zu einer Veränderung: Der Mittelwert hat sich von 26 auf 46 erhöht, die Standardabweichung sogar verdoppelt und die Sprachen samt Kombinationen sind von 393 auf 1161 gestiegen. Die Fehleinträge unter *Language* konnten bereits reduziert werden (2084 zu 1339), prozentual gesehen waren es vor dem Update 84,13 Prozent, nach dem Update nur noch 53,56 Prozent.

Vor dem Update:



Nach dem Update:



## 6.5 Datenvergleich: Herkunftsland

Unter den Herkunftsländern konnten vor allem Russland (142 auf 214) und Deutschland (9 auf 39) zulegen, aber auch Polen (31 auf 51) und Belarus (34 auf 52). Neu hinzu kam jetzt die USA, da hier vorher offenbar noch nicht genug Daten vorlagen. Offensichtlich müssen insbesondere bei den Herkunftsländern noch einige Daten ergänzt werden, dies zeigen vor allem die fehlenden Werte *missing values*. Von 90,02 Prozent (2229 missing values) fand hier eine Reduzierung auf 82,12 Prozent (2053 missing values) statt. Der Durchschnitt verringerte sich von 20 auf 14, die Standardabweichung zeigte nahezu keine Veränderung um nicht einmal ein halbes Prozent.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte BeschreibungVor dem Update:

Ein Bild, das Text, Screenshot, Schrift, Karte Menü enthält.

Automatisch generierte Beschreibung

Nach dem Update:

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Schrift, Dokument enthält.

Automatisch generierte Beschreibung

## 6.6 Datenvergleich: Systeme

Die Daten zu den Systemen lagen größtenteils vollständig vor, denn bei 2473 Systemen (vor dem Update) fehlten lediglich drei Einträge, nach dem Update fehlten 9 Systeme bei 2491 insgesamt. Mittelwert und Standardabweichung zeigten sich erwartbar mit ähnlichen Werten.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Diagramm, parallel enthält.

Automatisch generierte BeschreibungVor dem Update:

Ein Bild, das Text, Screenshot, parallel, Diagramm enthält.

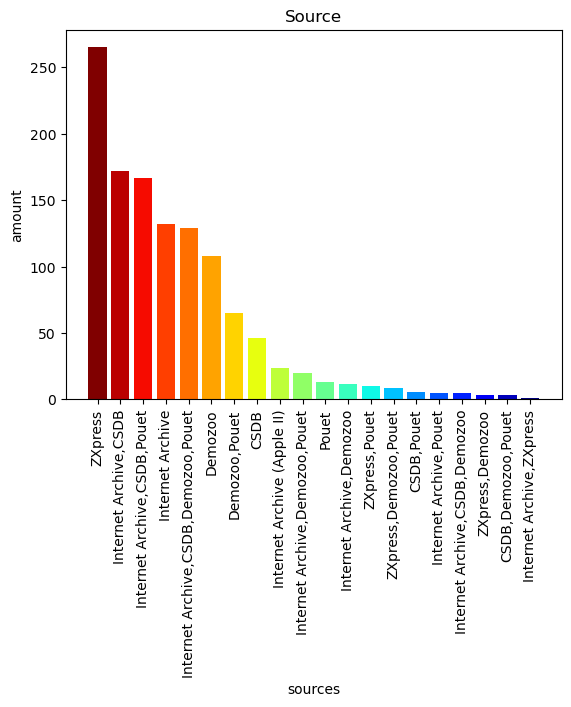
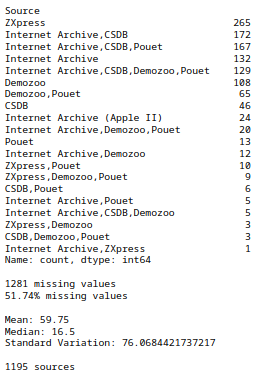
Automatisch generierte BeschreibungNach dem Update:

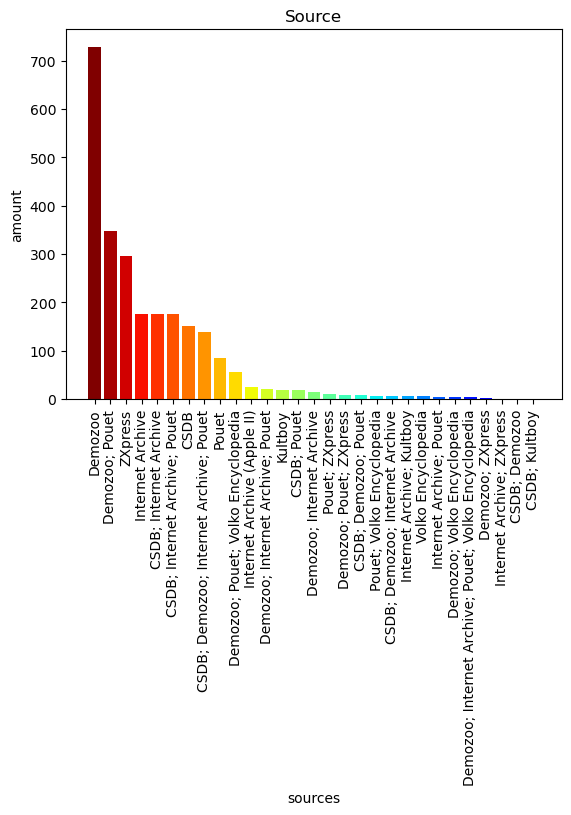
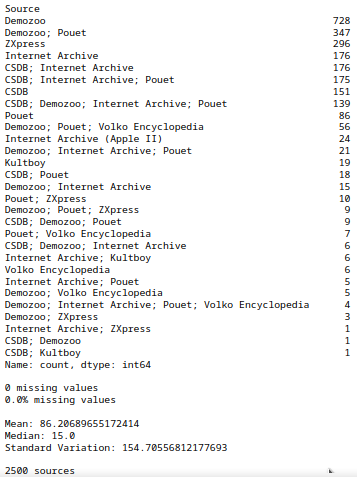
Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung:

## 6.7 Datenvergleich: Quellen

Von der Aktualisierung profitierte vor allem *Demozoo*. Vorher wurde *Demozoo* 108-mal als alleinige Quelle verzeichnet, nach dem Update 728-mal und damit ein neuer Spitzenwert unter allen genannten Quellen. Auch in Kombination mit *Pouet* kam es zu einem Sprung an zweite Stelle von 65 auf 347, der vorherige Spitzenreiter *ZXpress* rutschte auf Rang Drei ab (stieg von 265 auf 296). Überraschenderweise lagen vor dem Update noch 1281 missing values vor, nach dem Update gab es keinen einzigen Fehleintrag mehr. Der Mittelwert vor dem Update erhöhte sich von 59,75 auf 86,21, die Standardabweichung verdoppelte sich.

Vor dem Update:

Nach dem Update:

## 6.8 Korrelations-Matrizen

Vor dem Update wurde in der folgenden Grafik als *ALT* bezeichnet, nach dem Update als *NEU*. Viele Korrelationswerte sanken nach Update. Auffällig ist bereits bei *Magazine* (Anmerkung: im neueren Code als *Title* bezeichnet), dass es nahezu nicht mit anderen Kategorien korreliert, was an der hohen Anzahl der verschiedenen Magazine liegt. Die Werte vor dem Update weichen hier deutlicher voneinander ab. Bei *Language/Origin* war vorher noch eine hohe Übereinstimmung mit 0,808 zu verzeichnen, nachher nur noch 0,291, also ein sehr niedriger Wert. *Language/Systems* *und Language/Source* verzeichneten sogar deutlich höhere Werte nach dem Update. *Origin/Systems* und *Origin/Source* fielen jeweils um ca. 10% ab. *Systems/Source* ergab eine höhere Korrelation nachher. Die beste Übereinstimmung zeigen *Language* und *Source* mit 74,4 Prozent.

Ein Bild, das Text, Screenshot, Farbigkeit, Quadrat enthält.

Automatisch generierte Beschreibung

# 7. Probleme und Lösungen

## 7.1 Datenzugriff

Die Daten wurden zunächst nur von der Website diskmags.de als csv-Datei heruntergeladen, jedoch konnte nicht alles auf einmal heruntergeladen werden, sondern nur die jeweilige Anzahl der gelisteten Daten pro Webpage. Am Ende wurden von der Website vier verschiedene csv-Dateien heruntergeladen, die allerdings extra noch zu einer Datei zusammengefügt werden mussten. Dies war mit Python möglich, erforderte jedoch einen Mehraufwand, zudem waren die Daten der Website nicht aktuell. Die Lösung war der Download der gesamten csv-Daten aus Github. Im Vergleich zur Website gaben die Dateien auf Github zudem den aktuellen Stand der Datenlage wieder und es war kein extra Zusammenlegen der Daten nötig.

## 7.2 Das Datum im Pandas Dataframe

Das Problem mit dem Datum war, dass es nicht als solches erkannt wurde, sondern nur als nicht berechenbare Zahlenfolge. Nach langer Recherche konnte eine Umwandlungsfunktion gefunden werden. Pandas bietet hier die Funktion *.to\_datetime* an, die nachfolgend sowohl die Eingabe des umzuwandelnden Dataframes erfordert als auch des Datumsformats. Dank der jetzt ins Datumsformat transformierten Zellen konnten dadurch Berechnungen angestellt werden, um so die neue Spalte *Duration* zu kalkulieren bzw. einfügen zu lassen.

Ein Bild, das Text, Schrift, Screenshot, weiß enthält.

Automatisch generierte Beschreibung

## 7.3 Zeichnen von Lebensdauer-Abschnitten

Das folgende Beispiel befindet sich aufgrund eines Darstellungsfehlers nicht im Hauptteil, sollte aber aus Interesse trotzdem mitaufgenommen werden, weswegen in diesem Abschnitt *Probleme und Lösungen* eine Anmerkung gemacht wird. Blocklängen zeigen die Lebensdauer der Magazine in einer Grafik werden, allerdings gab es ein Problem mit der Werteumsetzung der y-Achse. Die Dauer in Jahren wurde als fünfstellige Zahl, statt einzeln angezeigt. Hier lag wohl irgendein Formatierungsproblem vor, der bis jetzt noch nicht gelöst werden konnte, dennoch ist hier schön der Unterschied zwischen den einzelnen Magazinen nach deren Lebensdauer zu erkennen:

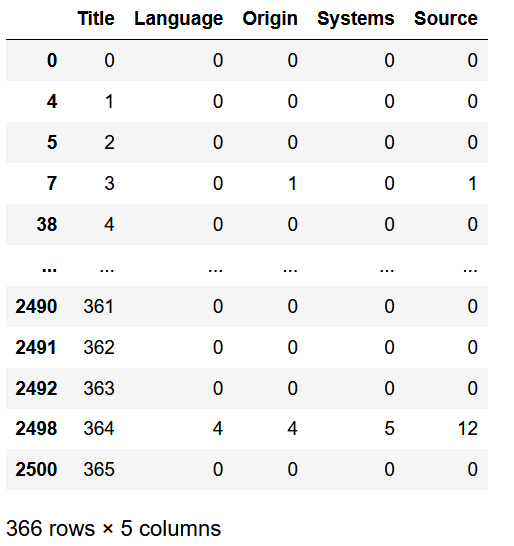
Ein Bild, das Text, Diagramm, Screenshot, Reihe enthält.

Automatisch generierte Beschreibung

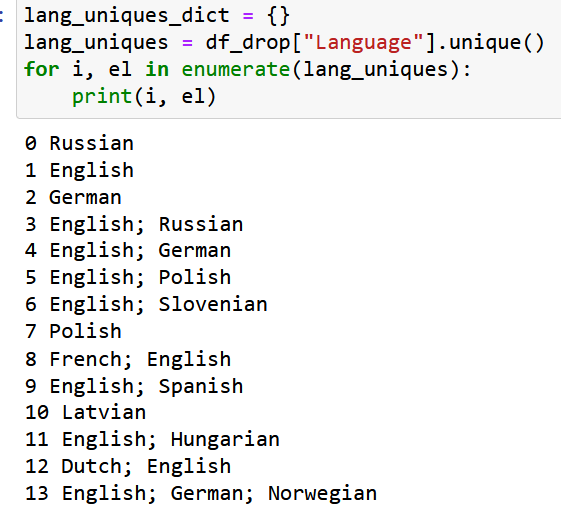
## 7.4 Umwandlung der Kategorien in Zahlenwerte für das maschinelle Lernen

Da es sich bei Kategorien wie *Title, Language, Origin* und *Systems* um Strings handelte, über maschinelles Lernen jedoch nur Zahlen verarbeitet werden konnten, war es nötig, die Kategorien in Zahlenwerte umzuwandeln. Die Funktion *pd.factorize()* in Pandas ermöglichte die Verarbeitung als Voraussetzung zum maschinellen Lernen. Im Screenshot direkt unter dem Code ist zu erkennen, dass jetzt Zahlen statt Text erscheint. Die Zahlen wurden der Reihe nach, also in aufsteigender Reihenfolge, eingefügt.





Um allerdings vor der User-Eingabe zu wissen, was eigentlich numerisch einzufügen ist (dass der entsprechende Name auch der Zahl entspricht), wurde über Code das Ganze nochmal entschlüsselt. In Klammern stand die entsprechende Kategorie (hier: *Language*), der Code-Aufbau blieb unverändert:



## 7.5 Zu lange Spalten-Namen in der CSV-Datei

Die Spaltenbezeichnungen waren in der originalen CSV-Datei zu lange, daher wurden die Namen entsprechend abgekürzt:

Original:

['Title', 'Magazine[Language]', 'Magazine[Origin]',

'Magazine[Start date]', 'Magazine[End date]', 'Magazine[Systems]',

'Magazine[Issues]', 'Magazine[Source]']

Änderung:

['Title', 'Language', 'Origin', 'Start Date', 'End Date', 'Systems',

'Issues', 'Source', 'Duration']

## 7.6 Korrelations-Score

Der zunächst verwendete *NMI -* *Normalized Mutual Information Score* (über sklearn importiert) wies bei den Korrelationswerten teilweise komplett andere Ergebnisse auf. Es stellte sich nach intensiver Recherche heraus, dass der *NMI* bei einem einzelnen Datenrahmen nicht viel Sinn macht, sondern eher darauf ausgelegt ist, mehrere Datenrahmen parallel zu vergleichen, was die klaren Abweichungen von der Konfusionsmatrix erklärt. Aus diesem Grund ist der *NMI* hier nicht aufgeführt, stattdessen liegt der Fokus auf der Konfusionsmatrix.

# 8. Literaturverzeichnis

* Dempster, Martin, and Donncha Hanna. *Statistik und Forschungsmethoden Für Psychologen und Sozialwissenschaftler Für Dummies*, John Wiley & Sons, Incorporated, 2019,S. 441.  
  *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/ub-wuerzburg/detail.action?docID=5782476.  
  Abgerufen am: 31.12.2023
* Hartmann, Florian; Kopp, Johannes; Lois, Daniel: Sozialwissenschaftliche Datenanalyse. Eine Einführung. Springer 2022, S.48/84.
* Julius-Maximilians-Universität Würzburg (Zentrum für Philologie und Digitalität): http://diskmags.de/index.php?title=Diskmag:Intro  
  Abgerufen am: 27.12.2023
* Jung, Alexander: Machine learning: The basics. (1st edition). Springer 2022, S.64.
* Khalusova, Maria: Machine Learning Model Evaluation Metrics part 2: Multi-class classification

https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/

Abgerufen am: 08.01.2024

* Noemi, Kurt: Stochastik für Informatiker. Eine Einführung in einheitlich strukturierten Lerneinheiten. Springer 2020, S.73.
* Roeder, Torsten (2022): Rescuing Diskmags: Towards Scholarly [Re]Digitisation of an Early Born-Digital Heritage. In: magazén – International Journal for Digital and Public Humanities. Vol 3 – Num. 1 – June 2022, S. 139-158.

Abgerufen am: 05.01.2024

* Wikipedia: https://de.wikipedia.org/wiki/Diskette  
  Abgerufen am: 27.12.2023

1. https://de.wikipedia.org/wiki/Diskette [↑](#footnote-ref-2)
2. http://diskmags.de/index.php?title=Diskmag:Intro [↑](#footnote-ref-3)
3. Roeder 2022, S. 140 [↑](#footnote-ref-4)
4. http://diskmags.de/index.php?title=Diskmag:Intro [↑](#footnote-ref-5)
5. Dempster/Donncha 2019, S.441 [↑](#footnote-ref-6)
6. Hartmann/Kopp/Lois 2022, S.48 [↑](#footnote-ref-7)
7. Noemi 2020, S.73 [↑](#footnote-ref-8)
8. Hartmann/Kopp/Lois 2022, S.84 [↑](#footnote-ref-9)
9. Jung 2022, S.64 [↑](#footnote-ref-10)
10. Khalusova 2022, https://www.mariakhalusova.com/posts/2019-04-17-ml-model-evaluation-metrics-p2/ [↑](#footnote-ref-11)