

# Time Awareness

Intelligente Wiedergabelisten für den Musikplayer Museek

Eine Gruppenarbeit von Etienne Geiser, Jestin Kannanmannil, Michael Grob  
in der Distributed Computing Group an der ETH Zürich, Sommer 2010  
Betreuer: Michael Kuhn, Samuel Welten

Zürich, 07. Oktober 2010

# Abstract

Durch die wachsende Grösse von privaten Musikbibliotheken werden neue Formen für die Auswahl und Suche von Liedern nötig.

In dieser Gruppenarbeit haben wir aus diesem Grund ein Tool erstellt, mit dessen Hilfe sich Musik entsprechend den Musikhörgewohnheiten des Benutzers abspielen lässt. Wir nehmen dabei an, dass dieses Hörverhalten je nach Zeitraum variiert. Aufgrund dieser Annahme lässt sich mit diesem Tool anhand des aufgezeichneten Musikhörverhaltens eine speziell auf den Benutzer und auf den Zeitpunkt abgestimmte Wiedergabeliste erstellen.

Um diese Annahme rechtfertigen zu können, haben wir eine Studie mit hunderten Musikhörern durchgeführt und ausgewertet.

# Inhaltsverzeichnis

Abstract.....	2
Inhaltsverzeichnis.....	3
Einführung .....	4
Grundlage .....	4
Ziel .....	4
Vorgehensweise .....	5
Datenloggen.....	6
Time Awareness basierte Wiedergabelisten .....	6
Anbindung an Facebook.....	9
Datenerhebung für Benutzerstudie .....	11
Auswertung.....	13
Klassifizierung über die Tageszeit.....	14
Klassifizierung über den Wochentag.....	17
Klassifizierung über Werktage und das Wochenende .....	19
Auswertung über die Zeit und den Wochentag.....	20
Fazit .....	25
Quellen.....	26
Anhang.....	27
Tageszeitauswertung .....	28
Wochentagauswertung.....	29
Werktag/Wochenende-Auswertung.....	37
Tageszeit/Wochentag-Auswertung.....	38

# Einführung

## Grundlage

Im Rahmen dieser Gruppenarbeit haben wir eine Erweiterung für das Projekt Musicexplorer entwickelt. Musicexplorer ist ein laufendes Forschungsprojekt an der Distributed Computing Group der ETH Zürich, das Musikbibliotheken in einem 32-dimensionalen Raum anordnet, in welchem die Distanz zweier Lieder umgekehrt proportional zu deren Ähnlichkeit ist. Aus dem Projekt ist das Musikabspielprogramm *museek* für Android-Mobiltelefone entstanden. Es bietet aufbauend auf der oben beschriebenen Anordnung von Musik verschiedene Features, die eigene Musikbibliothek zu durchsuchen. U. a. besteht die Möglichkeit, Musik anhand ihrer Ähnlichkeit abzuspielen, oder die Musikbibliothek kann in einer 2D/3D-Musiklandkarte durchsucht werden.

Da die privaten Musiksammlungen immer grösser werden und somit den Benutzer bei der Wahl eines Liedes überfordern, ist es mit Hilfe von unserer Erweiterung, die wir Time Awareness nennen, möglich, zu jedem Zeitpunkt ein für den Anwender passendes Lied zu finden.

Diese Erweiterung wird mit dem Android-SDK, der Entwicklungsumgebung für das Android-Betriebssystem, mit Eclipse in Java programmiert.

Um zu zeigen, dass diese Time Awareness ihren Zweck erfüllt, untersuchen wir, ob die Musikhörgewohnheiten von der Tageszeit und/oder vom Wochentag abhängig sind. Dadurch ist es uns mit unserer Erweiterung, die wir Time Awareness nennen, möglich, personalisierte Wiedergabelisten zu erstellen.

Die Resultate dieser Studie ergeben jedoch keine eindeutige Bestätigung für unsere Annahme, dass sich das Musikverhalten stark von der Zeit abhängt. Deshalb dürfte Time Awareness nur für eine kleine Anzahl Anwender interessant sein.

## Ziel

Wir bieten dem Benutzer personalisierte, zeitabhängige Wiedergabelisten an. Ausserdem überprüfen wir die Annahme, dass der bevorzugte Musikstil eines Nutzers von der Tageszeit und/oder dem Wochentag abhängig ist. Zu diesem Zweck führen wir eine Nutzerstudie durch.

# Vorgehensweise

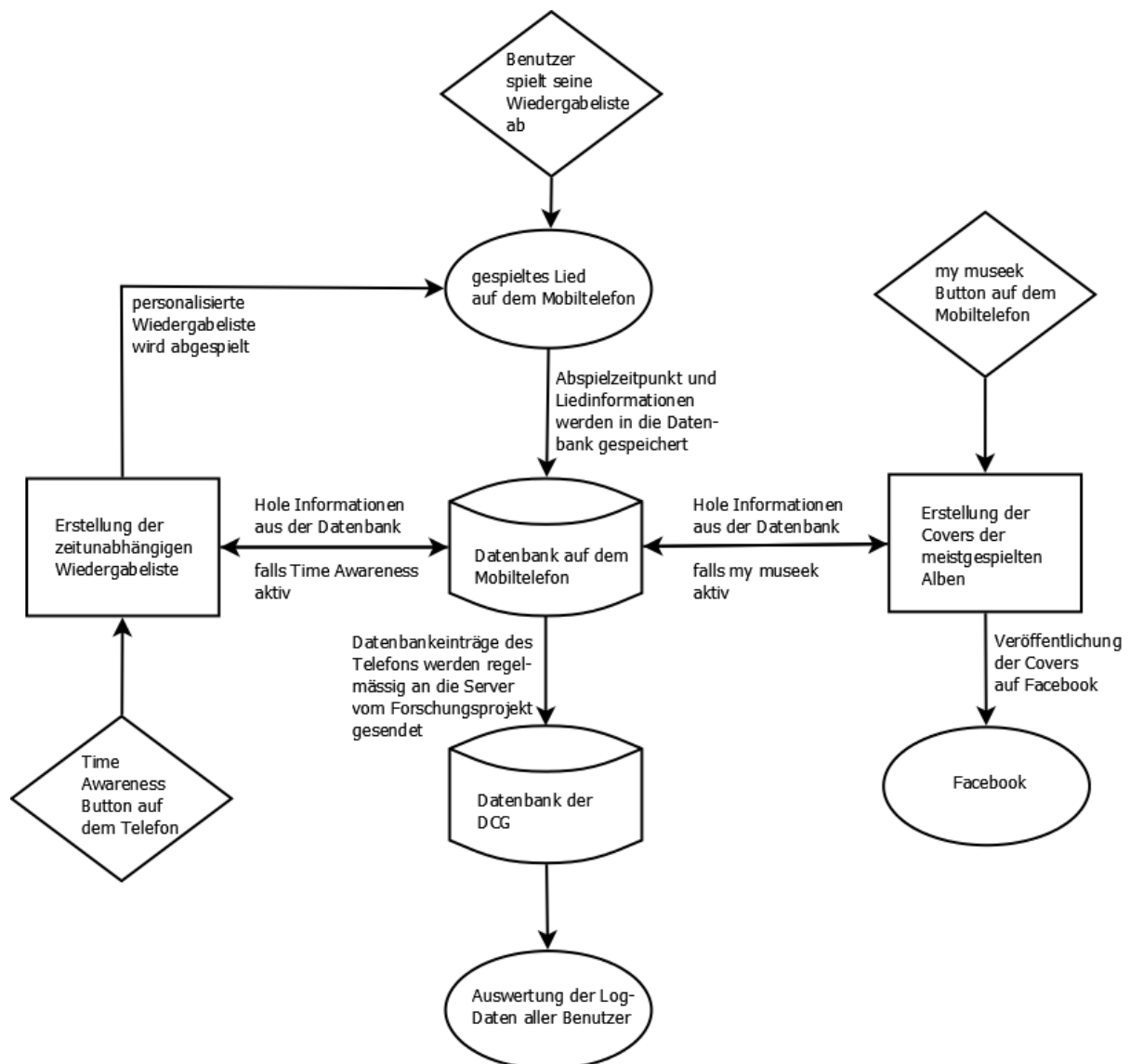


Abb. 1 Schema der Gesamtsystems (inkl. Testumgebung)

Jedes abspielte Lied auf *museek* wird in einer Datenbank geloggt, wie dies in der Abbildung 1 sichtbar ist.

Nach dem Erreichen einer Mindestanzahl von Einträgen (um eine gewisse Qualität der Wiedergabelistenberechnungen zu garantieren) wird dem Benutzer die Möglichkeit angeboten, Wiedergabelisten anhand dieser Einträge zu erstellen (Time Awareness). Entweder für die aktuelle Tageszeit/Wochentag-Kombination oder in erweiterten Einstellungen anhand (fast) beliebiger Kombinationen.

Als zusätzliches Feature wird dem Benutzer eine Anbindung an Facebook angeboten (*my museek*), welche seine Hörgewohnheiten (für Freunde) visualisiert. Der

Benutzer bestimmt ein Intervall (wahlweise 1 Tag, 1 Woche oder individuell bestimmte Intervalle), von dem er gerne die Favoriten seiner abgespielten Lieder veröffentlichen möchte. Aus den Log-Daten in diesem Intervall wird anschliessend eine Grafik aus Albumcovers erstellt und auf Facebook veröffentlicht.

Ausserdem werden die Log-Daten auf dem Mobiltelefon anonymisiert an die Server der DCG gesendet. Diese Daten werden für die Nutzerstudie benötigt.

### **Datenloggen**

Es werden eine Benutzer-ID, der Unix-Timestamp, der Wochentag, die Tageszeit (nur Stunde), ob das Lied vollständig abgespielt oder übersprungen wurde, ob die Time Awareness aktiviert war, sowie die ID und die Koordinaten des Liedes in einer SQL-Datenbank geloggt. Die Anzahl der Einträge in der Datenbank wird durch eine obere Schranke beschränkt. Damit soll ein zu hoher Speicherbedarf verhindert werden. Die Schranke darf aber auch nicht zu tief sein, damit die Log-Daten noch sinnvoll ausgewertet werden können.

Wie zuvor erwähnt werden die Einträge aus der SQL-Datenbank regelmässig an die Server des Forschungsprojekts gesendet. Mit diesen gewonnenen Daten können wir u. A. feststellen, wie viele Lieder ein Benutzer im Durchschnitt hört, indem wir die Gesamtzahl aller abgespielten Lieder durch die Anzahl aller Benutzer teilen.

Die Auswertung dieser Daten ergibt dabei durchschnittlich 1'600 geloggte Lieder in 3 Monaten. Der Speicherbedarf ist für diese Anzahl Log-Daten gering, daher setzen wir die obere Schranke auf 3'000 Log-Einträge. Das ergibt dann durchschnittlich Log-Einträge über die Zeitdauer eines halben Jahres, womit eine gute Auswertung möglich sein sollte.

Bei Erreichen dieser Schranke werden die ältesten Einträge gelöscht.

### **Time Awareness basierte Wiedergabelisten**

Eine Time Awareness basierte Wiedergabeliste ist eine Wiedergabeliste, die dem zeitlichen Hörverhalten des Benutzers angepasst ist. Sie soll Lieder enthalten, die der Benutzer (sehr wahrscheinlich) zu diesem Zeitpunkt hören möchte.

Wir haben eine untere Schranke an Log-Einträgen definiert, damit unsere Listen das Hörverhalten des Benutzers zuverlässig repräsentieren. Die untere Schranke wurde folgendermassen bestimmt: Als Ausgangspunkt nehmen wir die Tatsache, dass wir eine Wiedergabeliste für einen Wochentag UND eine Tageszeit erstellen

möchten. Als minimale Anforderung brauchen wir dazu zwei abgespielte Lieder für die gewünschte Tageszeit am gewünschten Wochentag. Die Auswertung von Benutzerdaten mit diesen Anforderungen ergibt eine durchschnittliche Anzahl Log-Einträge bis zum Erreichen der Anforderungen von 280.

Nach dem Erreichen der unteren Schranke erhält der Benutzer im Play-Mode-Menü einen neuen Button, der ihn die personalisierten Wiedergabelisten erstellen lässt.

Für das Erstellen der Wiedergabeliste werden ausgehend vom aktuellen Wochentag und der aktuellen Tageszeit die Datenbankeinträge untersucht:

Falls mehr als 20 Einträge für den Wochentag und die Tageszeit vorhanden sind, wird aus diesen die Wiedergabeliste erstellt. Falls nicht genügend Einträge gefunden werden, werden die Einträge zur entsprechenden Tageszeit gesucht (ohne den Wochentag einzuschränken). Falls auch diese Suche nicht genügend Einträge ergibt, werden alle Einträge zum entsprechenden Tag gesucht. Wenn diese Suche immer noch nicht zum gewünschten Ergebnis führt, werden die Einträge der ganzen Woche verwendet.

Die Wiedergabeliste wird dann automatisch entsprechend der gefundenen Einträge benannt.

Falls z.B. nur genügend Einträge für den Wochentag Samstag gefunden wurden, heisst die Liste *Saturday favorites*.

Um die Favoriten zu bestimmen, werden die Log-Einträge geclustert. D.h. die Einträge werden in  $k$  Gruppen (Cluster) aufgeteilt; in unserem Fall ist  $k$  gleich 5. Ein Cluster hat einen Schwerpunkt und zugewiesene Einträge.

Dazu verwenden wir den  $k$ -means Algorithmus mit einer zusätzlichen Vorkonditionierung.

Die Vorkonditionierung dient zur Erstbestimmung der Cluster. Damit lässt sich die zur Konvergenz des  $k$ -means Algorithmus nötige Anzahl Schritte deutlich verkleinern und die Rechenzeit lässt sich spürbar verkürzen.

(Man kann den  $k$ -means auch ohne Vorkonditionierung und nur mit zufällig gewählten Startpunkten verwenden.)

Der Abstand zwischen zwei Punkten ist die euklidische Norm der Koordinaten.

### Vorkonditionierung:

1. Aus allen Einträgen werden 250 zufällig ausgewählt (oder alle Einträge, falls weniger als 250 Einträge vorhanden sind). Es wird nur mit 250 Einträgen gerechnet, da sonst die Vorkonditionierung keine wesentliche Verkürzung der Rechenzeit mehr bewirkt.
2. Daraus wird zufällig ein Eintrag als erster Clusterschwerpunkt gewählt.
3. Von diesem Punkt wird der Abstand zu allen anderen Einträgen ausgerechnet.
4. Aus den verbleibenden Einträgen wird danach ein nächster Clusterschwerpunkt bestimmt. Dabei werden Einträge mit grösserem Abstand mit höherer Wahrscheinlichkeit gewählt.
5. Die Abstände der verbliebenen Punkte zu den Clusterschwerpunkten werden ausgerechnet und der Abstand zum jeweils am nächsten liegenden Clusterschwerpunkt wird für die Auswahl in Punkt 4 verwendet.
6. Die Schritte 4 und 5 werden solange wiederholt bis 5 Clusterschwerpunkte bestimmt sind.

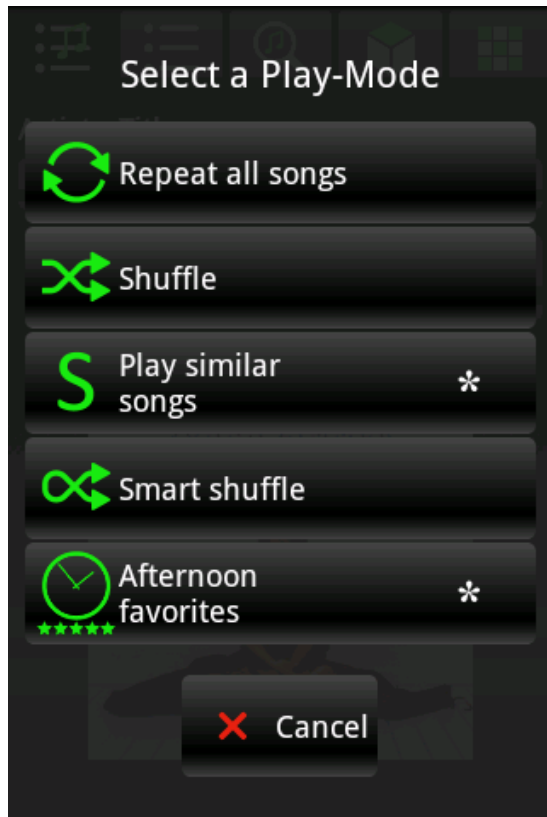
### k-means Algorithmus:

1. Die in der Vorkonditionierung berechneten 5 Clusterschwerpunkte dienen als Startpunkte für den k-means Algorithmus.
2. Alle Einträge werden dem jeweils am nächsten liegenden Clusterschwerpunkt zugewiesen und bilden mit diesem ein Cluster.
3. Von jedem Cluster wird der neue Schwerpunkt ausgerechnet.  
(Summe der Koordinaten geteilt durch die Anzahl zugewiesener Einträge)
4. Die Schritte 2 und 3 werden solange wiederholt bis die Änderung der Koordinaten der Clusterschwerpunkte innerhalb einer Toleranzgrenze liegen, jedoch maximal 15 Mal.

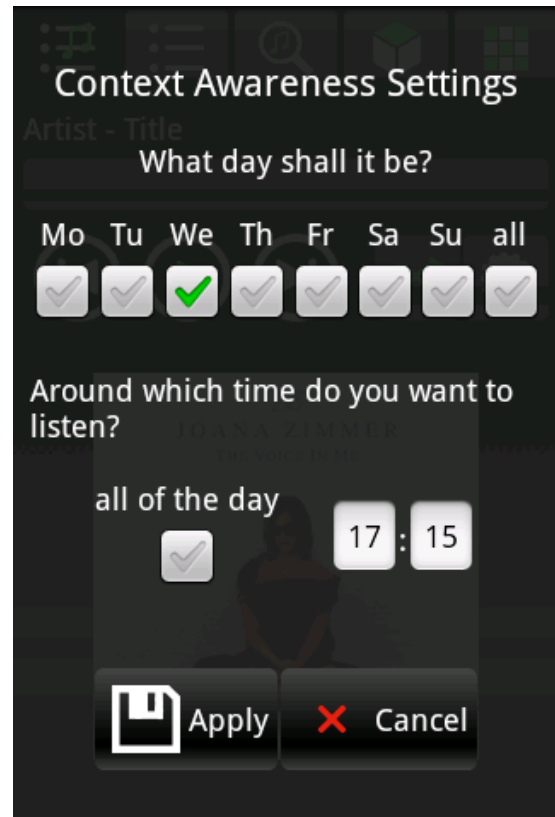
Aus den erhaltenen Clustern werden nun Lieder ausgewählt und der Wiedergabeliste beigefügt. Dabei wird ein Cluster abhängig von der Anzahl ihm zugewiesenen Einträgen ausgewählt (mehr Einträge -> grössere Wahlwahrscheinlichkeit). Vom Clusterschwerpunkt dieses Clusters werden nun mögliche Kandidaten mittels einer Funktion (*getNearestSongs*) von *museek* ermittelt. Danach wird verglichen



ob diese Kandidaten schon vor kurzem gespielt wurden; falls nicht wird dieser Kandidat zur Wiedergabeliste hinzugefügt.



**Abb. 2** Ansicht der Play Modus Auswahl mit dem neuen Button für eine Wiedergabeliste mit Nachmittags-Favoriten



**Abb. 3** Ansicht des erweiterten Menüs für den Context Awareness Modus

## Anbindung an Facebook

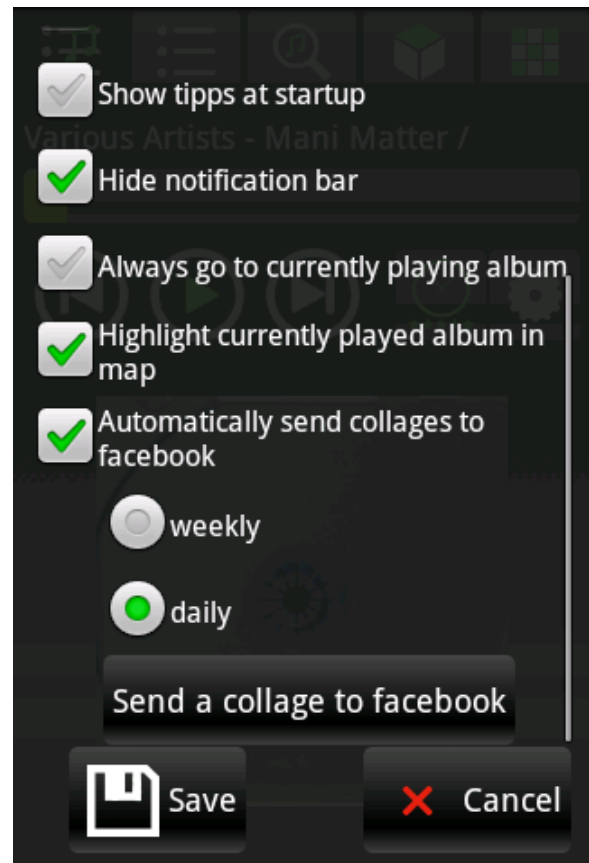
Um den aktuellen Musikgeschmack und die persönlichen Favoriten dem Umfeld eines Benutzers zugänglich zu machen, kann eine Collage der beliebtesten Alben eines Benutzers an Facebook gesendet werden. Dies kann täglich, wöchentlich oder manuell geschehen. Dazu werden in einem bestimmten Intervall (24 Stunden bei täglichem und manuellem, 7 Tage bei wöchentlichem Senden) alle entsprechenden Datenbankeinträge ausgewählt und nach der Häufigkeit der darin vorkommenden Alben sortiert. Dann werden die ersten fünf Alben ausgewählt und aus den dazugehörigen Albumcovern eine Collage zusammengestellt. Diese Collage wird danach auf Facebook im Account des Benutzers veröffentlicht.

Die Anbindung an Facebook geschieht über eine Facebook-Applikation, die der Anwender zu seinem Account hinzufügen kann und über die er dann die Collagen veröffentlichen kann.

Die Facebook-Applikation wurde unter dem Namen *my museek* und dem Logo in Abbildung 4 erstellt und speichert die Collagen in einem eigens erstellten Fotoalbum im Profil des Benutzers.



**Abb. 4** Das Logo für die Facebook-Applikation *my museek*



**Abb. 5** Das Einstellungs Menü mit den neuen Auswahlmöglichkeiten und dem Button für das manuelle Senden einer Collage

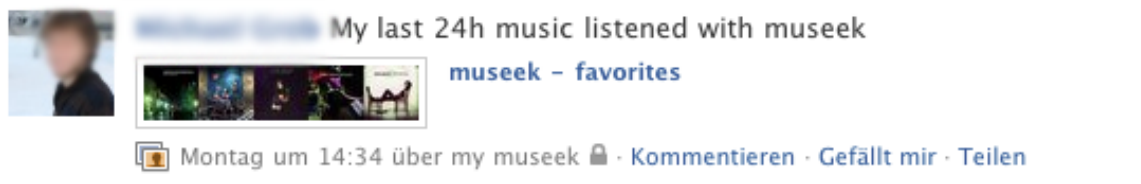
Die Anmeldung bei *my museek* und das Senden der Collagen geschehen über die offizielle API von Facebook.

Für die Anmeldung benötigt der Benutzer seine Facebook Login-Daten. Diese werden über HTTPS an den Facebook Login-Server geschickt und nicht auf dem Gerät gespeichert. Der Facebook Login-Server gibt als Antwort ein Zugriffs-Token zurück, mit dem dann die weiteren Anfragen an den Facebook Server autorisiert werden. Das Zugriffs-Token wird bei einem manuellen Senden nach erfolgreicher

Veröffentlichung der Collage gleich wieder deauthorisiert, so dass sich der Benutzer beim nächsten Senden von neuem einloggen muss. Beim automatischen Senden wird das Zugriffs-Token nicht gelöscht, damit das Einloggen bei jedem (automatischen) Sendevorgang entfällt. Die Lebensdauer des Token ist aber beschränkt (1 Monat), so dass zum Schutz vor Missbrauch nach dieser Zeit ein erneutes Einloggen des Benutzers nötig ist.



**Abb. 6** Collage aus Covers der fünf meistgespielten Alben in absteigender Reihenfolge



**Abb. 7** Veröffentlichung auf Facebook. Oben sieht man den Titel der Collage, neben der Collage steht der Name des Fotoalbums, in dem die Collagen gespeichert werden.

### Datenerhebung für Benutzerstudie

Nach der Fertigstellung des Programmteiles für das Loggen der abgespielten Lieder haben wir eine Funktion eingebaut, die diese Log-Daten zwecks Auswertung an einen Webserver schickt. Diese Funktion und das Loggen der Daten wurde in die aktuelle Version des *museek*-Players eingebaut und danach per Update den Benutzern zur Verfügung gestellt. Nach der Installation des Updates wird der Benutzer beim erstmaligen Gebrauch der neuen Version gefragt, ob er seine Log-Daten anonymisiert an den Webserver für die Auswertung schicken lassen will. Falls der Benutzer zustimmt, werden seine Log-Daten vor dem Beenden des Programms zu einem String zusammengefügt. Dieser wird, falls eine Internetverbindung besteht, via HTTP-POST an ein Skript auf dem Webserver des Instituts zur weiteren Bearbeitung gesendet. Falls keine Internetverbindung besteht wird das

Senden abgebrochen und beim nächsten Beenden des Programms ein neuer Versuch unternommen.

Das Skript nimmt den gesendeten String entgegen, trennt ihn in die einzelnen Einträge auf und speichert diese in einer MYSQL-Datenbank. Auf diese greifen wir dann für die Auswertung zu.

# Auswertung

Wir überprüfen nun, ob sich die anfangs gestellte These, dass sich der Musikgeschmack und der bevorzugte Musikstil eines Benutzers je nach Tageszeit und Wochentag ändert. Dazu werden wir die Daten aus der Benutzerstudie in den gewünschten Intervallen nach nachfolgenden Kriterien auswerten.

Die Auswertung haben wir am 24.09.2010 ausgeführt.

hash	timestamp	timeZoneOff	hour	day	artistId	titleId	playMode	skipped
-9f2530b320b14b4751d1aecb77bd0ae	1277676106816	3	0	1	575507	0	5	1
-9f2530b320b14b4751d1aecb77bd0ae	1277676109207	3	0	1	3803	0	5	1
5bb0e3d83e4a7e2c758b4e005820b734	1277699905973	0	6	1	694	51903	5	1
57da244b109031fa3ea317c06ce66192	1277675271817	4	23	7	3083	9527	4	0
57da244b109031fa3ea317c06ce66192	1277675514742	4	23	7	166	32111	4	0
57da244b109031fa3ea317c06ce66192	1277675699845	4	23	7	214	6296	4	0
57da244b109031fa3ea317c06ce66192	1277675910362	4	23	7	399	5277	4	0
57da244b109031fa3ea317c06ce66192	1277676105882	4	0	1	233	6861	4	0
57da244b109031fa3ea317c06ce66192	1277676293507	4	0	1	3062	8530372	4	0
57da244b109031fa3ea317c06ce66192	1277676579621	4	0	1	2398	84190	4	0
57da244b109031fa3ea317c06ce66192	1277676813340	4	0	1	2278	16271	4	0
57da244b109031fa3ea317c06ce66192	1277677006769	4	0	1	2278	32931	4	0
57da244b109031fa3ea317c06ce66192	1277677210045	4	0	1	2278	6464	4	0
57da244b109031fa3ea317c06ce66192	1277677417988	4	0	1	2278	23036	4	0

**Abb. 8** Ein Ausschnitt aus der Datenbank mit den aufgezeichneten Benutzerdaten

Im Detail untersuchen wir, ob in einem Intervall gehörte Lieder näher beieinander liegen (also ähnlicher sind) als die Lieder aller betrachteten Intervalle.

Die Qualität der aufgezeichneten Benutzerdaten ist stark schwankend. Mit nachfolgend genannten Kriterien werden die Benutzer mit unbrauchbaren Daten soweit als möglich aussortiert und gewisse Grundvoraussetzungen an Benutzer mit auswertbaren Daten gesetzt. So muss z.B. ein Benutzer mindestens 100 nicht übersprungene Log-Einträge haben, d.h. er muss eine gewisse Zeit geloggt haben, bevor er in die Auswertung einbezogen wird.

Generelle Kriterien, damit ein Benutzer in die Auswertung einbezogen wird, sind:

- Mindestens 100 Log-Einträge
- Die Zeitdauer, während der ein Benutzer Lieder geloggt hat, geteilt durch die Anzahl Lieder muss grösser sein als 10 Minuten.

$$\text{Durchschnittliche Dauer eines geloggtten Liedes} = \frac{\text{letzter Timestamp} - \text{frühester Timestamp}}{\text{Anzahl gehörter, nicht übersprungener Lieder}} > 10 \text{ Minuten}$$

Das bedeutet, dass ein Benutzer theoretisch während der ganzen Zeit, in der er Lieder geloggt hat, ununterbrochen Lieder mit der Dauer von jeweils 10 Minuten gehört haben könnte. D.h. dieses Kriterium soll ein möglichst natürliches Hörverhalten (zeitliche Lücken im Log-Verlauf) erzwingen.

- Der Benutzer steht nicht auf der schwarzen Liste. Die schwarze Liste beinhaltet Benutzer, deren Log-Daten eine Anomalie aufweisen. Z.B. alle Koordinaten aller Log-Daten sind exakt gleich.

Benutzer mit offensichtlich falsch eingestellter Systemuhr werden aussortiert. D.h. Benutzer mit einem Log-Startdatum oder Log-Enddatum, das in der Zukunft liegt (später als 2 Tage nach dem Zeitpunkt der Auswertung) oder zu weit in der Vergangenheit (früher als 01.06.2010: ab diesem Zeitpunkt war die neue Version des Programms mit der Log-Funktion als Update verfügbar).

Es werden nur Log-Einträge von Liedern, die bis zum Ende gehört wurden, berücksichtigt, d.h. ‚geskippte‘ Lieder werden ignoriert.

### **Klassifizierung über die Tageszeit**

Der Tag wird in folgende Intervalle aufgeteilt:

- Intervall 0 (Nacht): 00:00 – 05:59
- Intervall 1 (Morgen): 06:00 – 11:59
- Intervall 2 (Nachmittag): 12:00 – 17:59
- Intervall 3 (Abend): 18:00 – 23:59

Folgende Kriterien sind nötig, damit ein Benutzer in die Auswertung über die Tageszeit einbezogen wird:

- Mindestens 10 Log-Einträge in einem Zeitintervall  
Dieses Kriterium bedingt eine gewisse Anzahl an abgespielten Liedern in einem 6-Stunden-Intervall. 10 Lieder à durchschnittlich 3.5 Minuten ergeben eine Zeitdauer von etwas über 30 Minuten, während derer ein Benutzer in einem Intervall Musik gehört haben muss.

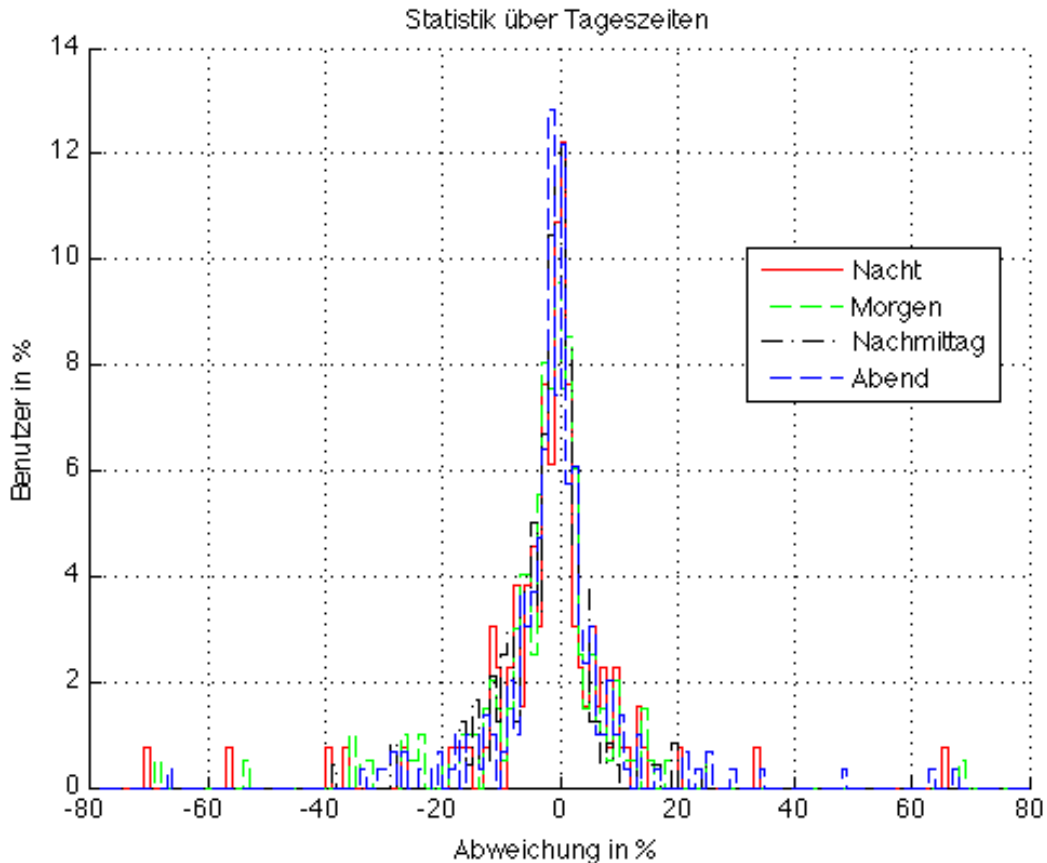
## Auswertung

- Mindestens 6 gleiche Zeitintervalle an verschiedenen Tagen  
Damit gleichen wir eventuelle Vorlieben an einem bestimmten Wochentag aus, indem wir zusätzliche Zeitintervalle an verschiedenen Tagen verlangen.
- Mindestens 2 verschiedene Zeitintervalle, die die obengenannten Kriterien erfüllen

Die Auswertung der Benutzerdaten nach obengenannten Kriterien ergibt 351 gültige Benutzer. In nachfolgender Tabelle sind die erhaltenen Werte für die mittlere prozentuale Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder des jeweiligen Benutzers ersichtlich.

Type of statistics		Time					
# songs:	278'529						
# users:	351						
	# users	# songs	mean over divergences	variance	users with negativ divergence (in %)	users with divergence smaller than -15% (in %)	users with divergence bigger than 15% (in %)
night	131	45'535	0.025	6.44	56.49	6.87	3.05
morning	199	61'815	-3.186	1.85	61.31	9.04	2.01
afternoon	239	86'743	-1.119	1.28	59.83	2.92	2.51
evening	296	84'436	-1.167	1.57	56.08	6.76	3.72

**Tab. 1** Prozentuale Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder eines Benutzers (Tageszeit)



**Fig. 1** Auswertung der Tageszeit über 351 Benutzer mit insgesamt 278'529 Liedern

Die X-Achse zeigt die durchschnittliche Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder des jeweiligen Benutzers.

Alle Werte aller ausgewerteten Benutzer werden schliesslich in Intervallen vom Minimum bis zum Maximum der durchschnittlichen Abweichung in Schritten von 1 summiert und in der Graphik eingetragen.

Auf der Y-Achse lässt sich nun der prozentuale Anteil der ausgewerteten Benutzer mit einer Abweichung in einem bestimmten Bereich ablesen.

Als Beispiel:

In obiger Graphik haben etwa 25% der Benutzer eine Abweichung vom Mittelwert der Abstände all ihrer ausgewerteten Lieder um den Bereich von 0%.

Die Abweichungen sind über die einzelnen Intervalle sehr gering. Trotz 351 Benutzern lässt sich nur am Morgen eine Tendenz in Richtung negativer Werte, also kleineren Abständen der Lieder, feststellen.



Am Nachmittag und am Abend ist zwar der Mittelwert negativ, aber die Varianz sehr gross, daher lässt sich keine klare Aussage machen.

In der Nacht bewegen sich die meisten Werte um 0% herum, wiederum mit einer sehr grossen Varianz. Dieses Intervall hat jedoch am wenigsten Benutzer, daher wird sich dieser Wert bei einer erneuten Auswertung mit mehr Benutzern noch verändern.

### **Klassifizierung über den Wochentag**

Folgende Kriterien sind nötig, damit ein Benutzer in die Auswertung über den Wochentag einbezogen wird:

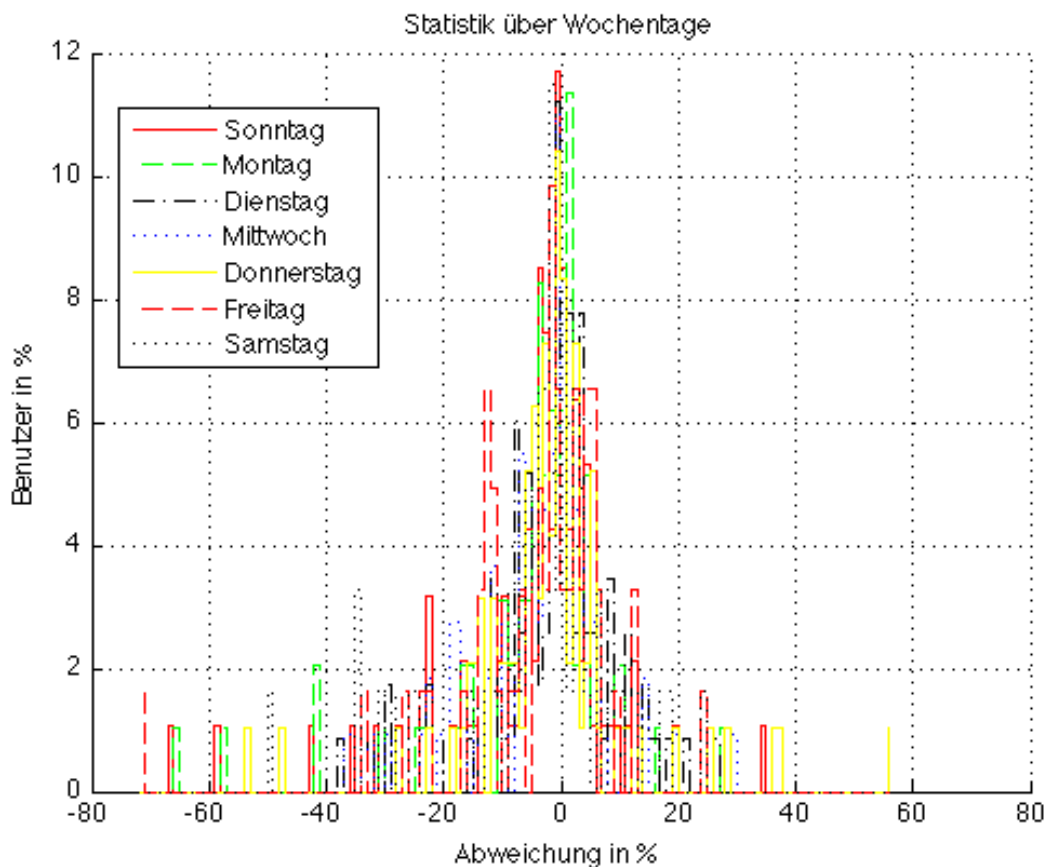
- Mindestens 35 Log-Einträge an einem Wochentag  
Das ergibt bei einer durchschnittlichen Lieddauer von 3.5 Minuten knapp über 2 Stunden, während derer ein Benutzer an einem Tag Musik gehört haben muss.
- Mindestens 2 gleiche Tage, die obiges Kriterium erfüllen  
Ein Benutzer muss an mindestens zwei gleichen Wochentagen jeweils mindestens 35 Lieder gehört haben, damit erreichen wir eine Mittelung der Daten über die Wochen.
- Mindestens 3 verschiedene Tage, die die obengenannten Kriterien erfüllen

Die Auswertung ergibt 150 Benutzer. In nachfolgender Tabelle sind die erhaltenen Werte für die mittlere prozentuale Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder des jeweiligen Benutzers ersichtlich.

## Auswertung

Type of statistics		Day					
# songs:	141'462						
# users:	150						
	# users	# songs	mean over divergences	variance	users with negativ divergence (in %)	users with divergence smaller than -15% (in %)	users with divergence bigger than 15% (in %)
sunday	94	18'868	-4.378	1.83	63.82	13.82	2.13
monday	97	22'074	-4.376	1.70	61.85	12.37	3.09
tuesday	116	24'287	-1.80	1.02	56.034	7.76	4.31
wednesday	109	23'282	-4.58	1.23	66.97	16.51	2.75
thursday	96	23'301	-1.89	1.89	62.5	8.33	6.25
friday	61	15'376	-4.130	1.72	60.65	9.84	1.64
saturday	61	14'274	-3.893	1.64	62.30	13.11	4.92

**Tab. 2** Prozentuale Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder eines Benutzers (Wochentag)



**Fig. 2** Auswertung der Wochentage über 150 Benutzer mit insgesamt 141'462 Liedern (Auswertung für jeden einzelnen Tag: siehe Anhang)

Mit Ausnahme des Dienstags und des Donnerstags ist die mittlere Abweichung bei allen Tagen im Bereich zwischen -3.8 und -4.6 Prozent bei einer nahezu halb so grossen Varianz. Das bedeutet, dass sich an diesen Tagen unsere Annahme relativ gut bestätigt. Jedoch werfen die Werte von Dienstag und Donnerstag Zweifel an der Aussagekraft unserer Auswertung auf, da sie so stark von den anderen abweichen. Diese Abweichung könnte auf einen Fehler in der Herangehensweise der Auswertung hinweisen.

### Klassifizierung über Werktag und das Wochenende

Die Aufteilung in Werktag und Wochenende ist nicht Bestandteil der angebotenen Wiedergabelisten in *museek*. Wir möchten aber herausfinden, ob diese Aufteilung eine bessere Klassifizierung ermöglicht. Wir nehmen an, dass sich Arbeitszeit und Freizeit auch in der Musikauswahl unterscheiden und sich somit ein Unterschied zwischen der gehörten Musik während der Arbeitswoche und dem Wochenende ergibt.

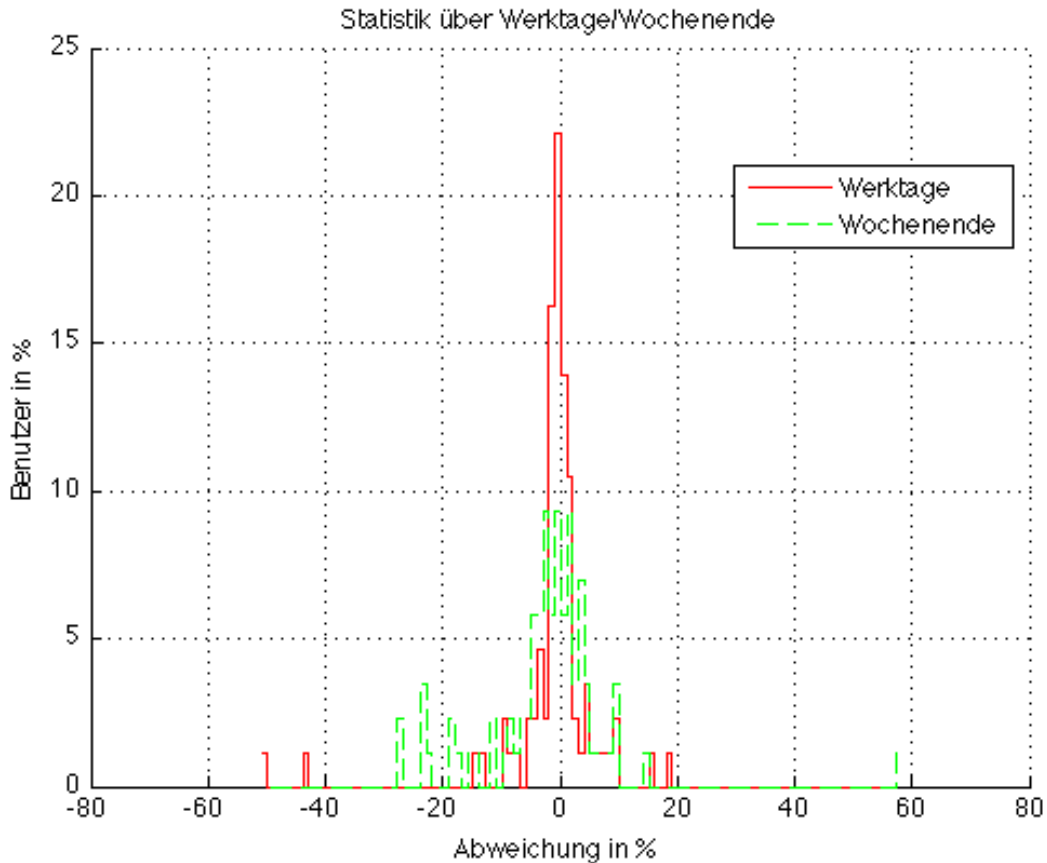
Folgende Kriterien sind nötig, damit ein Benutzer in die Auswertung über die Werktag/Wochenenden einbezogen wird:

- Mindestens 60 Log-Einträge an einem oder mehreren Werktagen und gleich viele an einem oder mehreren Wochenendtagen in der gleichen Kalenderwoche
- Mindestens 2 Wochen, in denen das Kriterium in Punkt 1 erfüllt wird

Die Auswertung ergibt 86 Benutzer. In nachfolgender Tabelle sind die erhaltenen Werte für die mittlere prozentuale Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder des jeweiligen Benutzers ersichtlich.

Type of statistics	Weekend/Workday						
# songs:	76'767						
# users:	86						
	# users	# songs	mean over divergences	variance	users with negativ divergence (in %)	users with divergence smaller than -15% (in %)	users with divergence bigger than 15% (in %)
workday	86	49'901	-1.045	0.74	53.48	2.32	2.32
weekend	86	26'866	-2.137	1,14	59.30	11.62	1,16

**Tab. 3** Prozentuale Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder eines Benutzers (Werktag/Wochenende)



**Fig. 3** Auswertung der Werktag/Wochenenden über 86 Benutzer mit insgesamt 76'767 Liedern

Bei der Aufteilung in Werktag und Wochenenden liegen beide Mittelwerte im negativen Bereich bei einer geringen Varianz (0.5-mal so gross). Allerdings ist die Anzahl Benutzer mit 86 noch sehr gering. Es lässt sich aber eine Tendenz in Richtung Bestätigung unserer Annahme feststellen.

An den Werktagen ist die Abweichung bei allen Benutzern sehr gering; so haben nur gerade etwas über 2% der Benutzer eine Abweichung grösser als 15%.

### **Auswertung über die Zeit und den Wochentag**

Dies ist eine Kombination aus den Auswertungen über die Zeit und den Wochentag. Und diese ist für die Auswertung unserer Annahmen sehr interessant, da wir für jede Time-Awareness basierte Wiedergabeliste zuerst nach diesen Kriterien versuchen die Musik auszuwählen.

Folgende Kriterien sind nötig, damit ein Benutzer in die Auswertung über die Zeit und den Wochentag einbezogen wird:

- Mindestens 35 Log-Einträge an einem Wochentag

## Auswertung

- Mindestens 2 gleiche Wochentage die obiges Kriterium erfüllen
- Mindestens 3 verschiedene Wochentage, die obengenannte Kriterien erfüllen
- Mindestens 10 Log-Einträge in einem Zeitintervall.
- Mindestens 6 gleiche Zeitintervalle an verschiedenen Tagen
- Mindestens 2 Zeitintervalle, die obengenannte Kriterien erfüllen

Die Zeitintervalle sind gleich wie in der Auswertung über die Zeit.

Die Auswertung ergibt 68 Benutzer. In nachfolgender Tabelle sind die erhaltenen Werte für die mittlere prozentuale Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder des jeweiligen Benutzers ersichtlich.

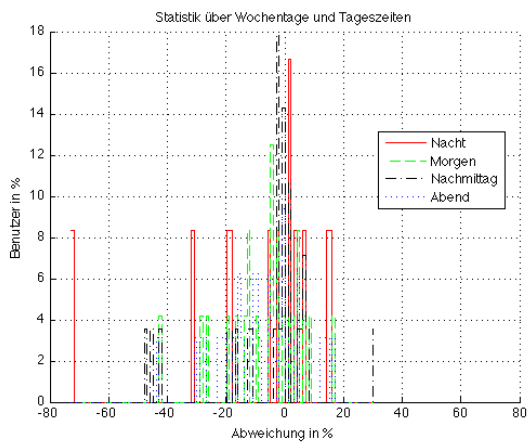
Type of statistics:		Day and Time						
# songs:		83'809						
# users:		68						
		# users	# songs	mean over divergences	variance	users with negativ divergence (in %)	users with divergence smaller than -15% (in %)	users with divergence bigger than 15% (in %)
sunday	night	12	1'840	-8.775	5.54	50.00	33.33	8.33
	morning	24	2'243	-6.736	1.64	70.83	20.83	4.17
	afternoon	28	3'637	-5.845	2.72	60.71	21.43	3.57
	evening	32	2'917	-7.906	1.93	68.75	31.25	3.13
monday	night	15	2'176	-5.771	1.87	53.33	26.67	6.67
	morning	35	3'309	-6.604	4.09	65.71	31.43	5.71
	afternoon	36	4'606	-7.262	3.00	58.33	25.00	5.56
	evening	39	3'494	-6.155	3.64	61.54	23.08	5.13
tuesday	night	18	2'573	-8.235	1.80	72.22	33.33	0.00
	morning	37	3'284	-6.919	1.42	72.97	18.92	5.41
	afternoon	37	4'374	-1.526	1.51	51.35	13.51	10.81
	evening	32	3'323	-6.918	3.01	62.50	25.00	6.25
wednesday	night	15	2'158	-7.403	2.31	60.00	20.00	6.67
	morning	34	3'625	-3.777	2.09	64.71	14.71	8.82
	afternoon	39	5'025	-5.074	1.92	69.23	12.82	7.69
	evening	35	3'580	-8.342	3.91	68.57	22.86	5.71

## Auswertung

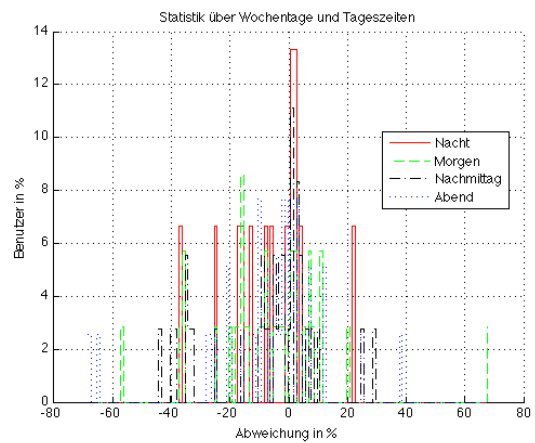
thursday	night	16	3'077	-13.059	2.51	81.25	31.25	0.00
	morning	38	4'177	-3.483	4.42	65.79	18.42	7.89
	afternoon	34	4'948	-10.937	4.75	82.35	29.41	5.88
	evening	26	2'689	-8.158	2.78	69.23	23.08	3.85
friday	night	12	1'755	-5.397	2.52	58.33	25.00	8.33
	morning	11	1'864	-15.303	6.55	63.64	45.45	9.09
	afternoon	15	3'148	-8.426	4.57	60.00	26.67	6.67
	evening	11	1'425	-6.502	2.93	63.64	18.18	9.09
saturday	night	16	1'783	-13.425	10.72	81.25	37.50	6.25
	morning	12	2'198	-6.505	1.00	75.00	16.67	0.00
	afternoon	14	1'785	-6.031	1.67	42.86	28.57	0.00
	evening	11	1'374	-7.013	0.64	81.82	18.18	0.00

**Tab. 4** Prozentuale Abweichung vom Mittelwert der Abstände aller ausgewählten Lieder eines Benutzers (Tageszeit/Wochentag)

Sonntag:

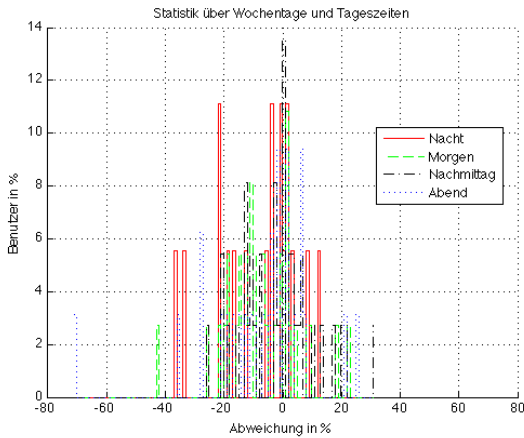


Montag:

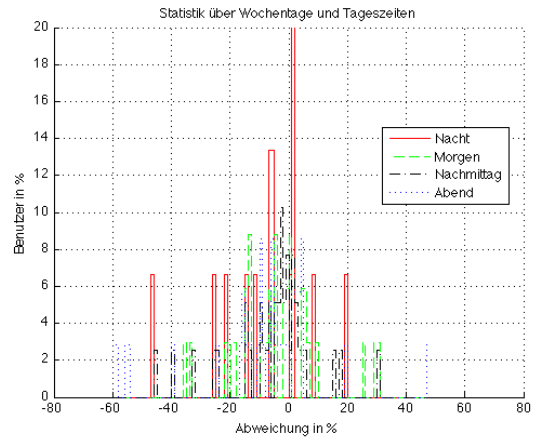


# Auswertung

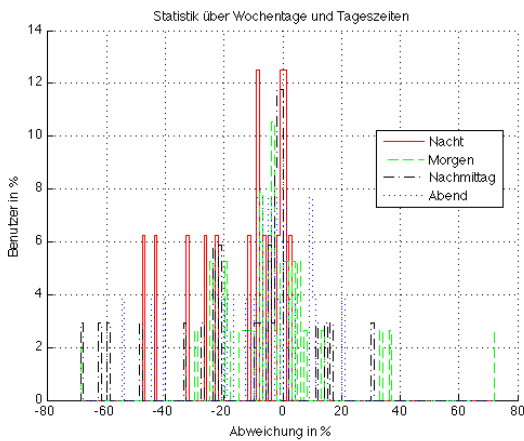
Dienstag:



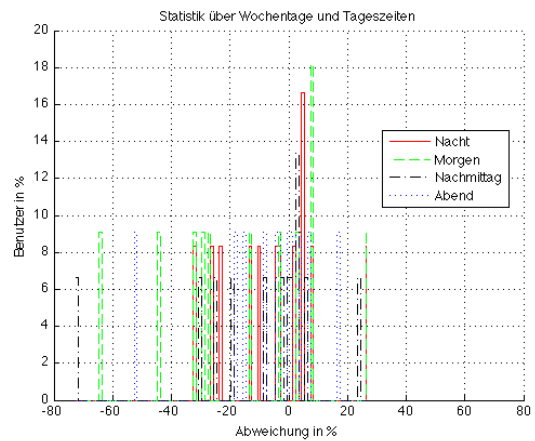
Mittwoch:



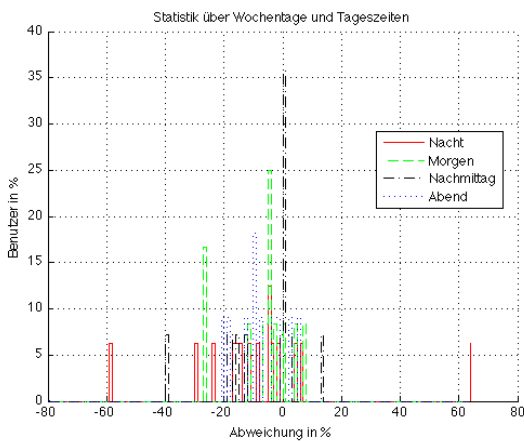
Donnerstag:



Freitag:



Samstag:



**Fig. 4** Auswertung über die Zeit und über den Wochentag über 68 Benutzer mit insgesamt 83'809 Liedern

Die Auswertung über die Zeit und die Wochentage liefert in der Mehrzahl der Intervalle ein gutes bis sehr gutes (z. B. Samstagabend) Ergebnis.

## Auswertung

---

Allerdings ist die Benutzerzahl pro Intervall stark schwankend. In manchen Intervallen wird die Auswertung lediglich mit knapp über 10 Benutzern durchgeführt, was natürlich keine statistisch relevante Aussage und Auswertung zulässt.

Die niedrige Benutzerzahl liegt an den hohen Anforderungen an die Intervalle (Bedingungen für Tage UND Tageszeiten müssen von einem Benutzer erfüllt werden) und daher auch an der relativ kurzen Zeit, in der die Benutzerstudie durchgeführt wurde.



## Fazit

Anhand der ausgewerteten Daten lässt sich die Frage, ob die von jeweils einem Benutzer gehörte Musik über einzelne Intervalle ähnlicher ist als über ein grösseres Intervall, nicht eindeutig beantworten. Es lässt sich eine Tendenz in positiver Richtung feststellen, die aber durch Auswertungen von weiteren Daten und vor allem mit einer grösseren Datenmenge noch bestätigt werden müsste.

Die Auswertung über die Wochentage und vor allem die Auswertung über die Wochentage und die Tageszeiten liefern gute Ansätze und sind es sicher wert genauer verfolgt zu werden.

Die Unterteilung in Tageszeiten bestätigt unsere zu Beginn getroffene Annahme nicht, da nur eine leichte Verschiebung in Richtung negative Werte festgestellt werden kann.

Die von uns gewählten Kriterien schränken die Anzahl Benutzer zu fest ein, so dass ein einzelner Anwender einen zu grossen Einfluss auf den Durchschnitt ausübt.

Deshalb ist die Aussagekraft der Auswertung nicht zu stark. Jedoch sehen wir keine Anzeichen dafür, dass unsere Annahmen bezüglich des Hörverhaltens bestätigt werden. Das Fehlen eines solchen Hörverhaltens macht die Time Awareness Erweiterung für die meisten Benutzer obsolet.

Die Anbindung an Facebook ist unabhängig von der Auswertung der Benutzerdaten und fordert nur eine geringe Anzahl Log-Einträge pro Tag oder Woche.

Diese Erweiterung ist eine gute Möglichkeit, Werbung für *museek* zu machen und das Programm vor allem Facebook Anwendern, d.h. einem breiten Benutzerkreis, durch virale Verteilung bekannt zu machen.

# Quellen

<http://developer.android.com/>

<http://www.apache.org/>

<http://developers.facebook.com/docs/>

<http://github.com/facebook/facebook-android-sdk.git>

<http://en.wikipedia.org/wiki/K-means++>

[http://en.wikipedia.org/wiki/K-means\\_clustering](http://en.wikipedia.org/wiki/K-means_clustering)

[http://home.dei.polimi.it/matteucc/Clustering/tutorial\\_html/kmeans.html](http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html)

# Anhang

## **Allgemeine Auswertung der Benutzerdaten:**

Anzahl aller geloggtten Lieder: 2'097'977

Anzahl aller ganz angehörten Lieder: 1'128'529

Anzahl Benutzer: 12'167

Anzahl Benutzer, die in der Auswertung berücksichtigt werden: 2'200

## **Tageszeit-Auswertung:**

Anzahl gültiger Benutzer: 351

Anzahl gültiger Lieder: 278'529

## **Wochentag-Auswertung:**

Anzahl gültiger Benutzer: 150

Anzahl gültiger Lieder: 141'462

## **Werktag/Wochenend-Auswertung:**

Anzahl gültiger Benutzer: 86

Anzahl gültiger Lieder: 76'767

## **Tageszeit/Wochentag-Auswertung:**

Anzahl gültiger Benutzer: 68

Anzahl gültiger Lieder: 83'809

## Tageszeitauswertung

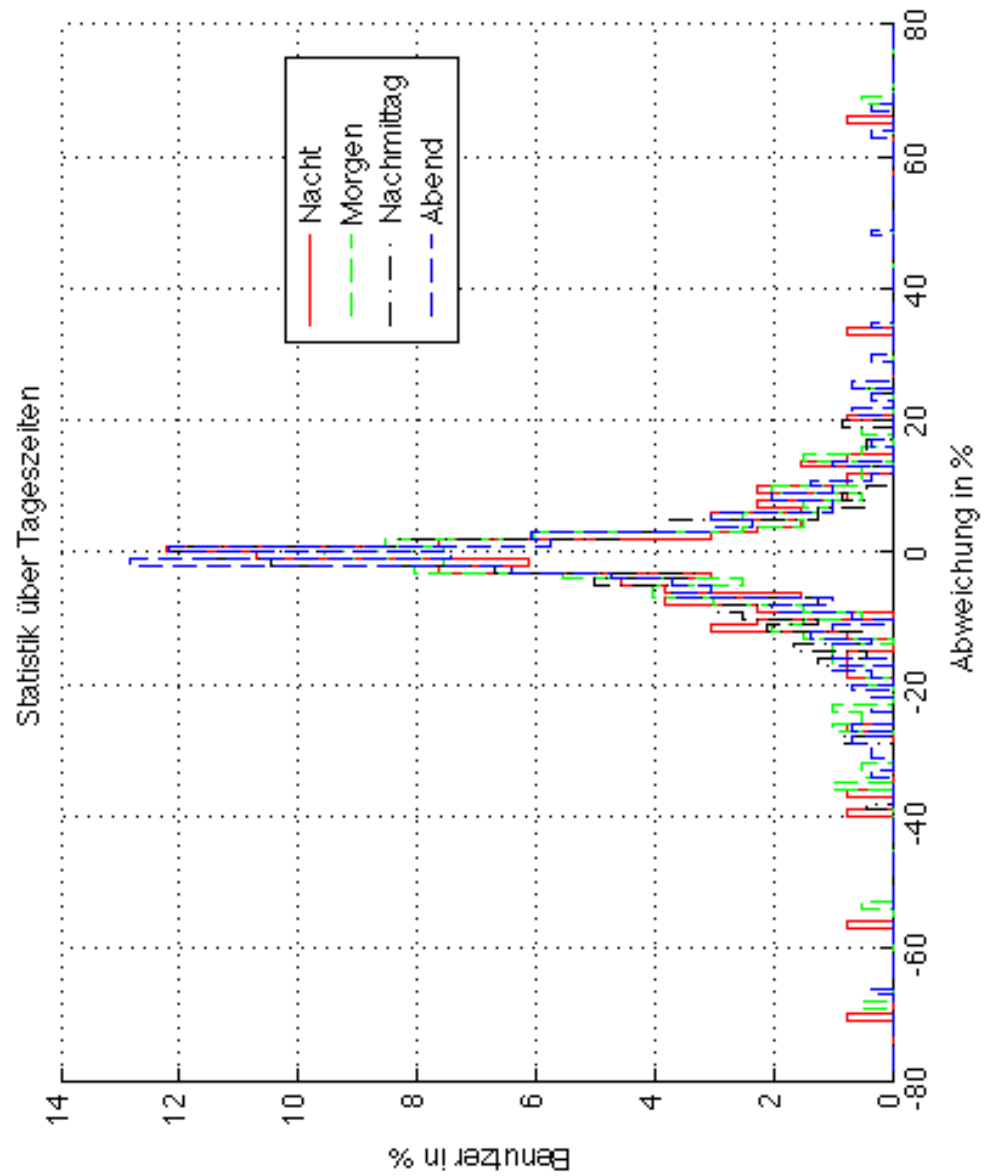


Fig. 5 Auswertung der Zeit über alle Benutzer

## Wochentagsauswertung

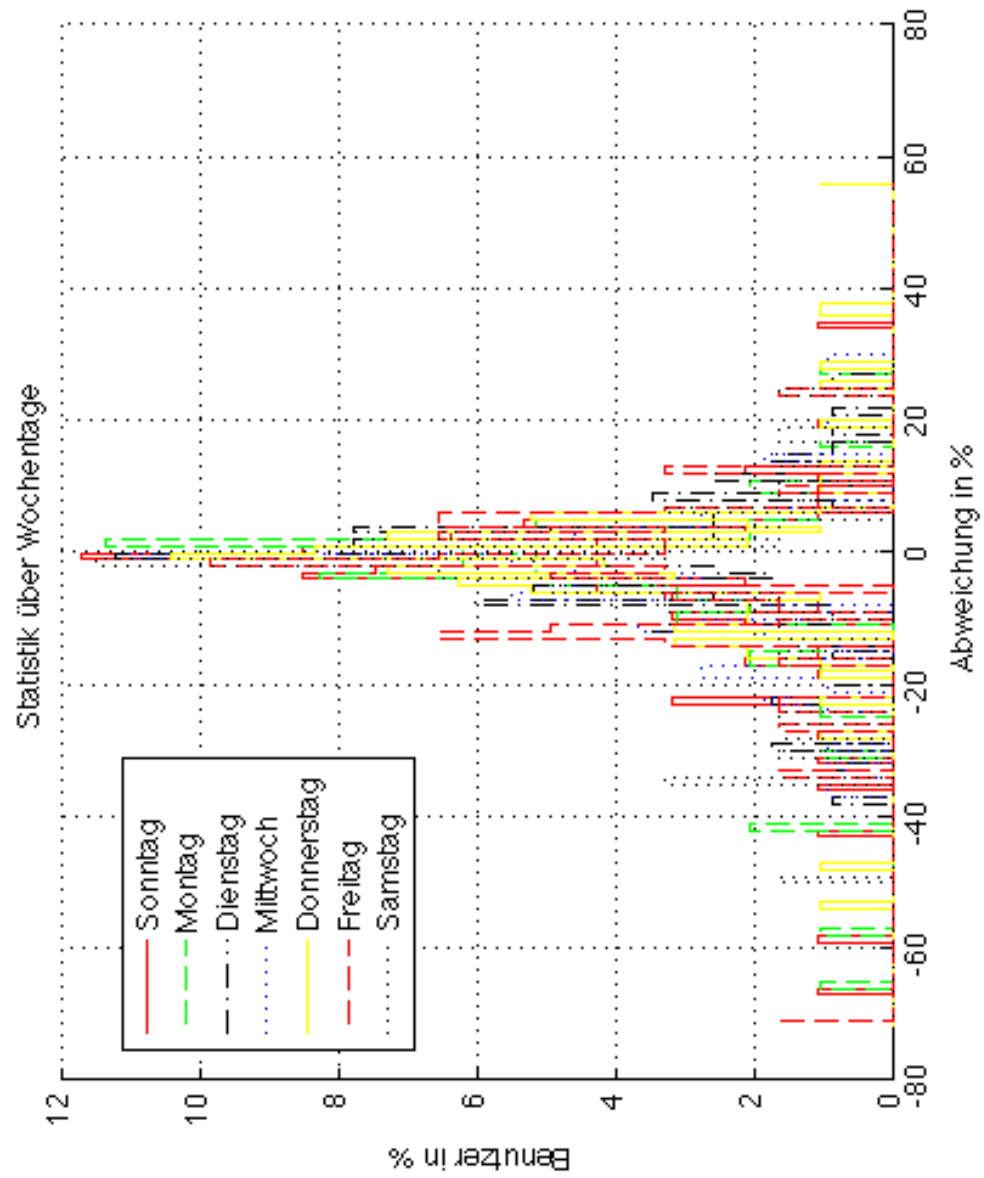


Fig. 6 Auswertung der Wochentage über alle Benutzer

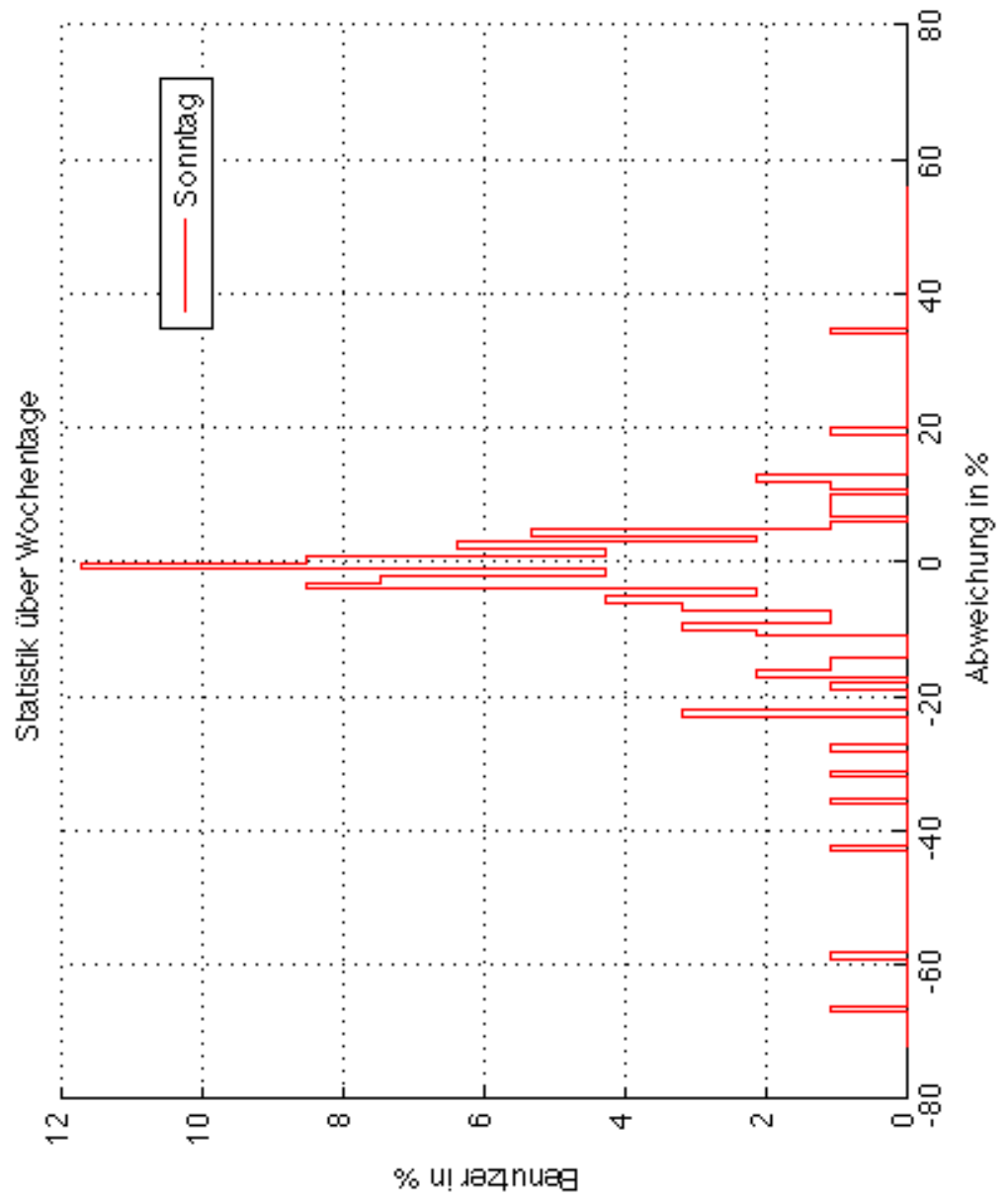


Fig. 7 Auswertung der Sonntage über alle Benutzer

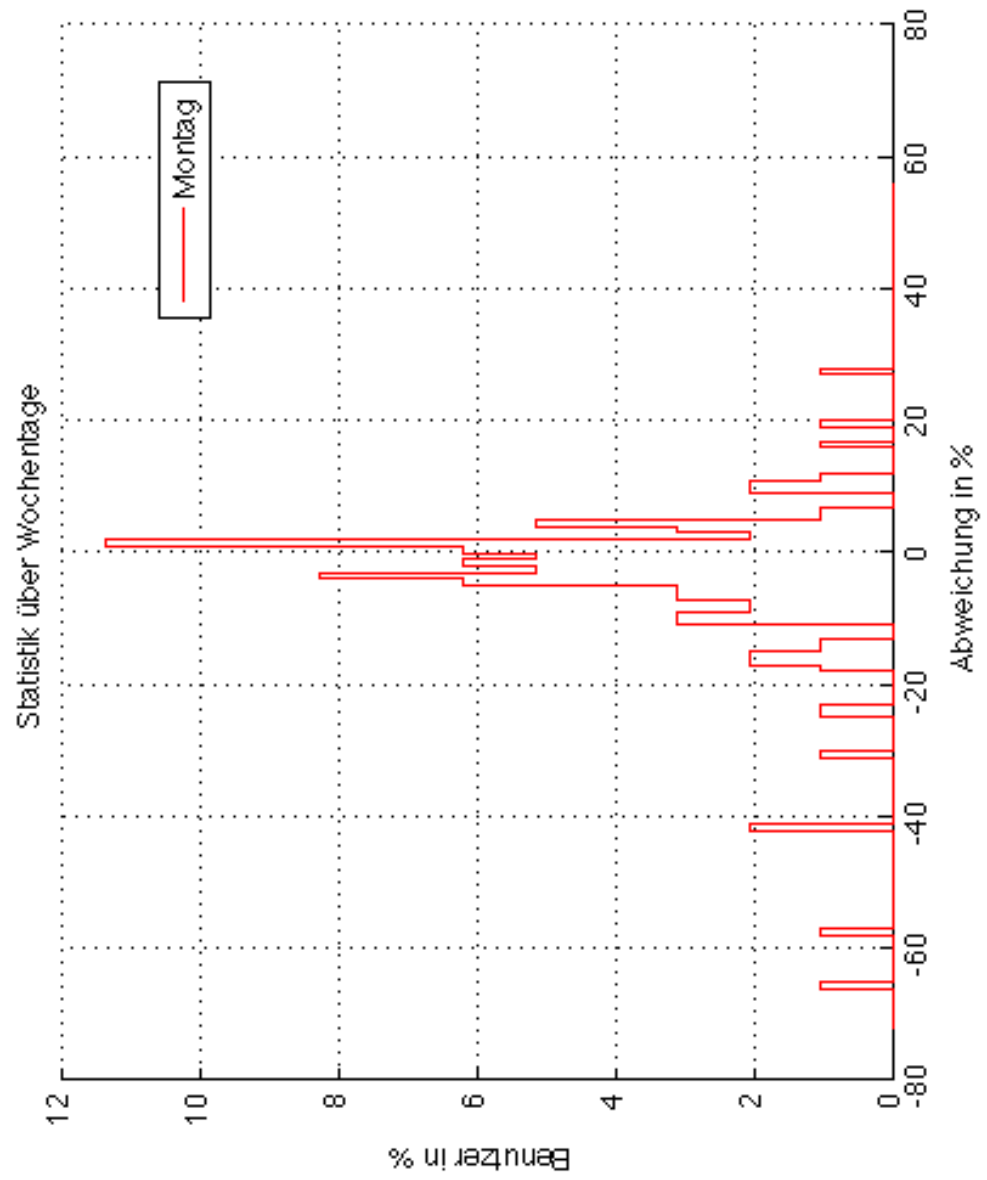


Fig. 8 Auswertung der Montage über alle Benutzer

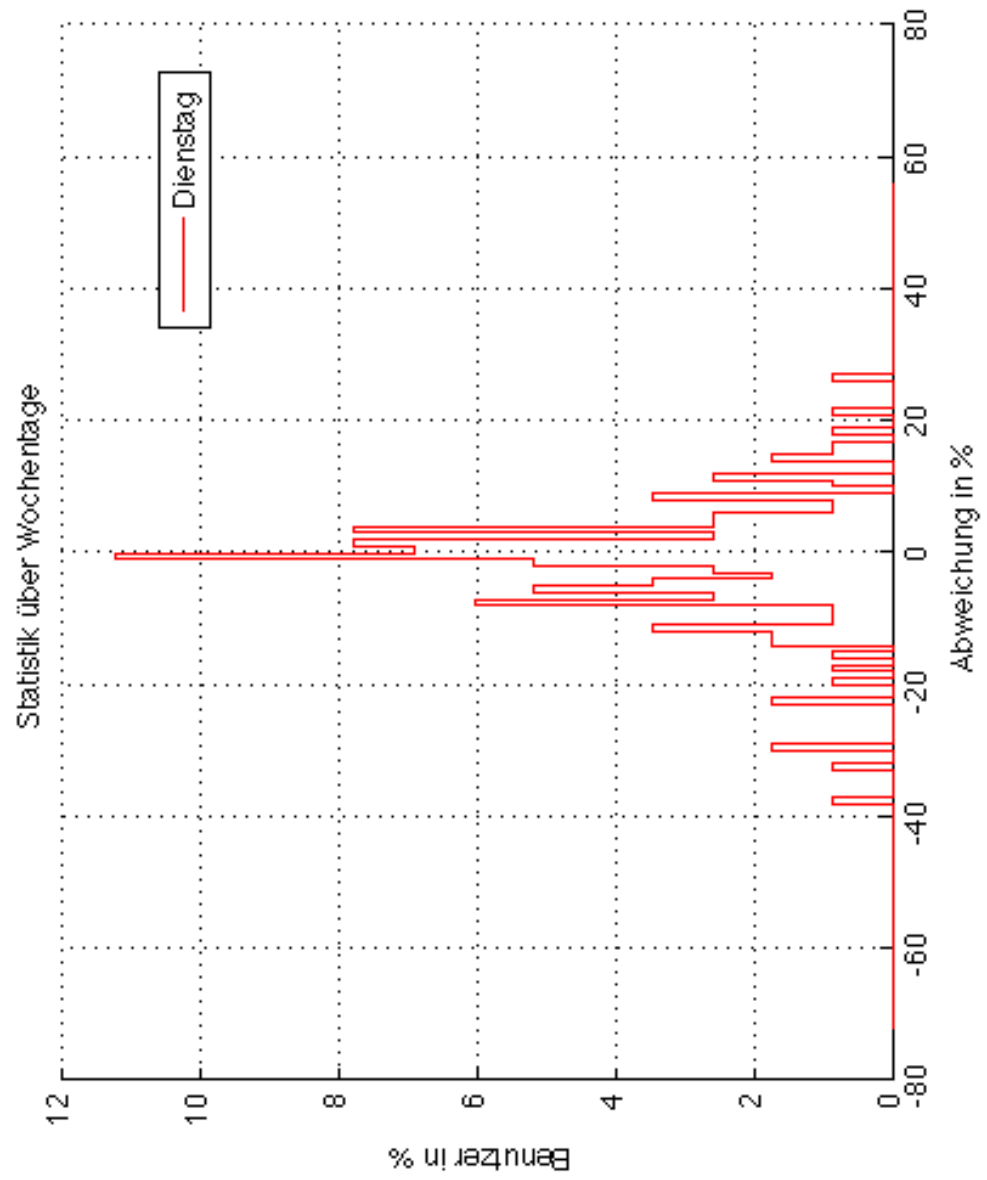


Fig. 9 Auswertung der Dienstage über alle Benutzer



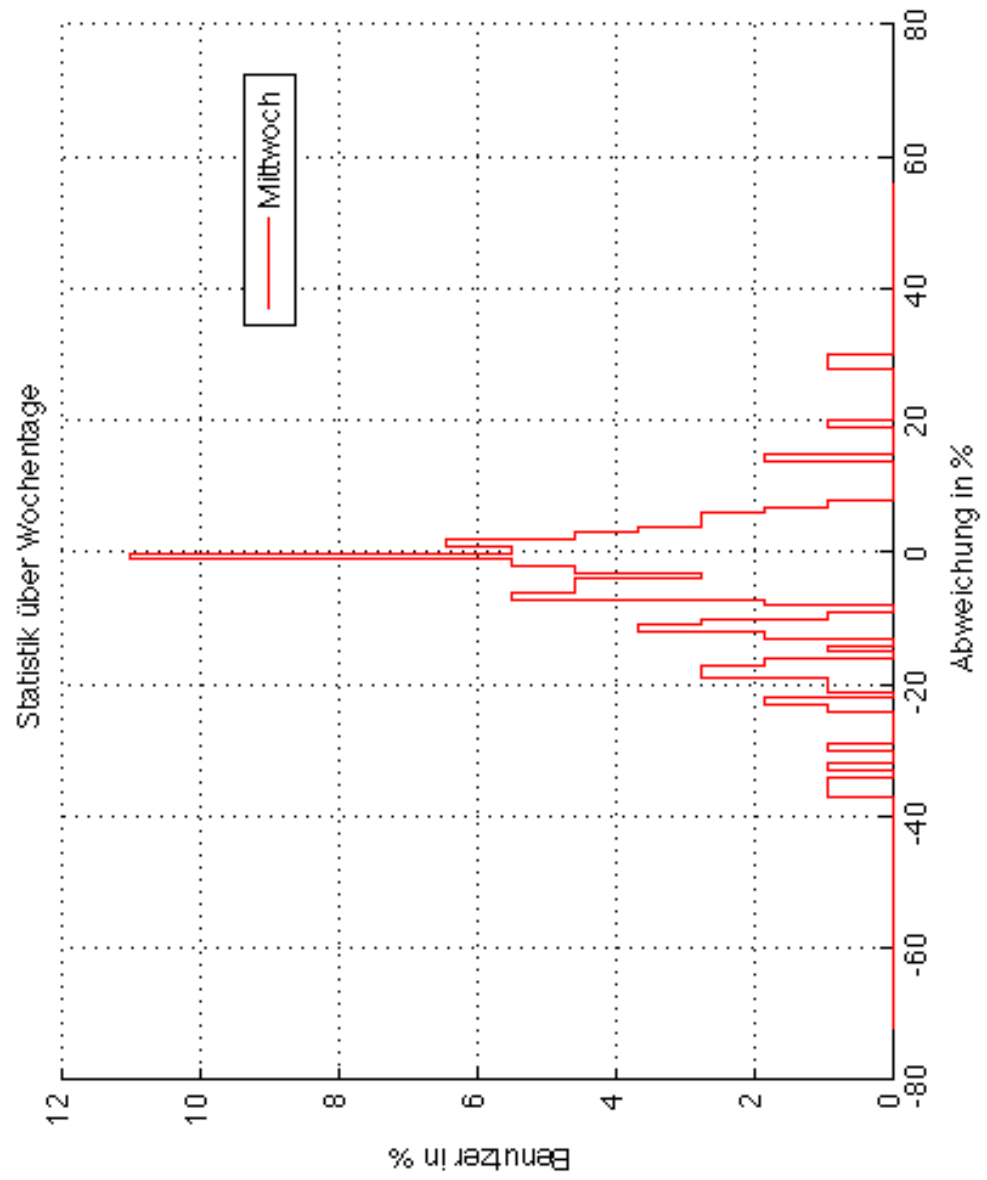


Fig. 10 Auswertung der Mittwoche über alle Benutzer

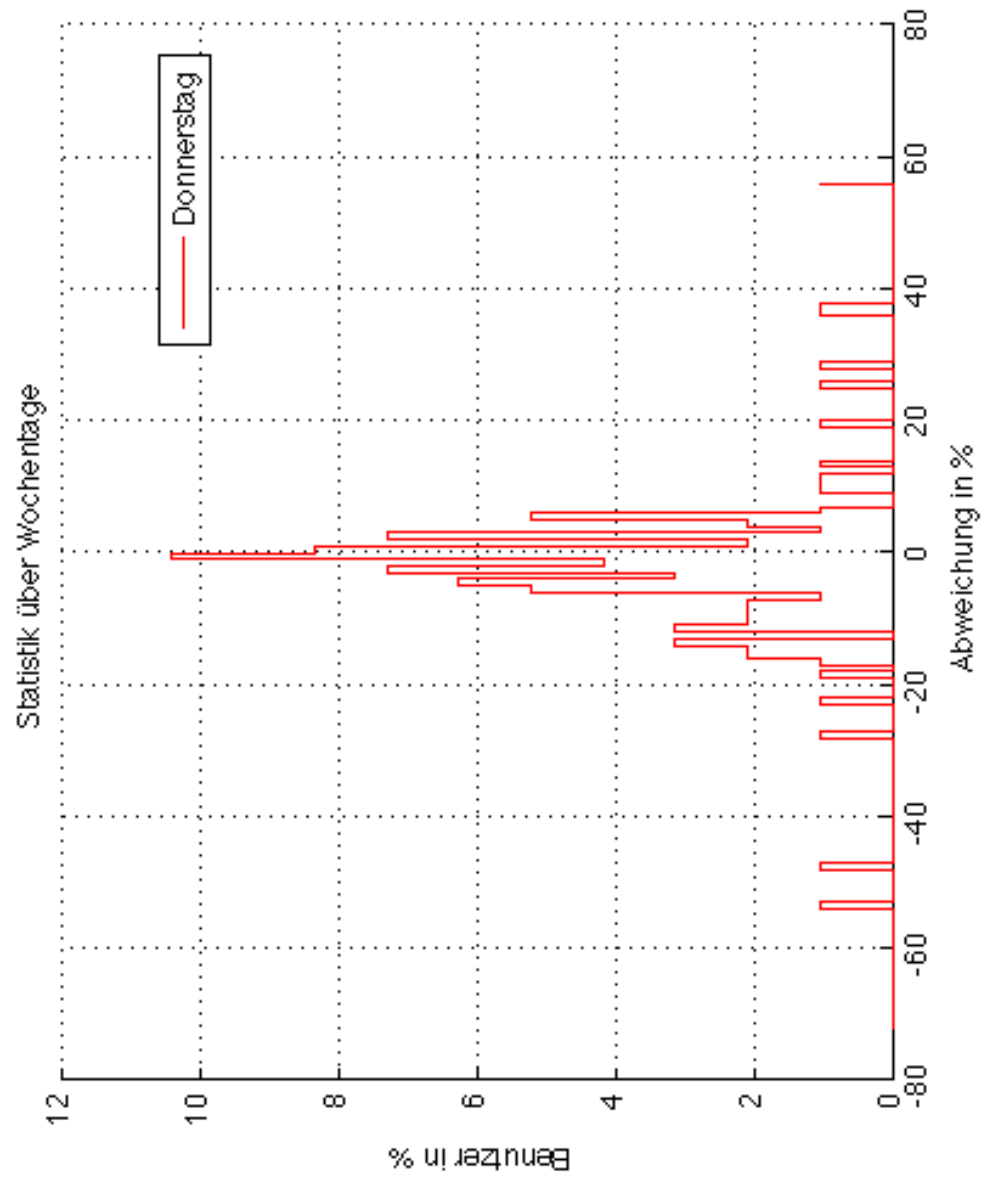


Fig. 11 Auswertung der Donnerstage über alle Benutzer

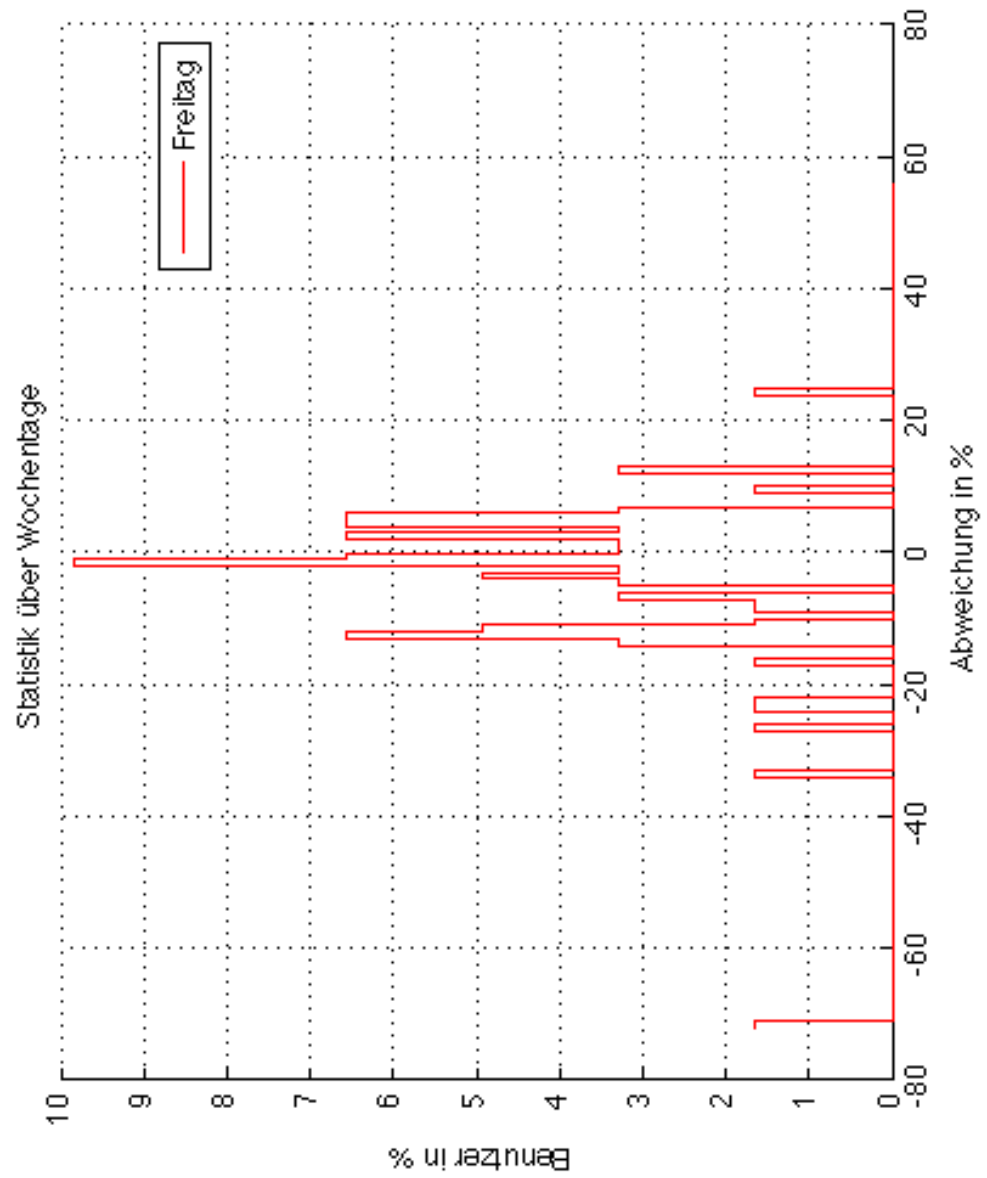


Fig. 12 Auswertung der Freitage über alle Benutzer

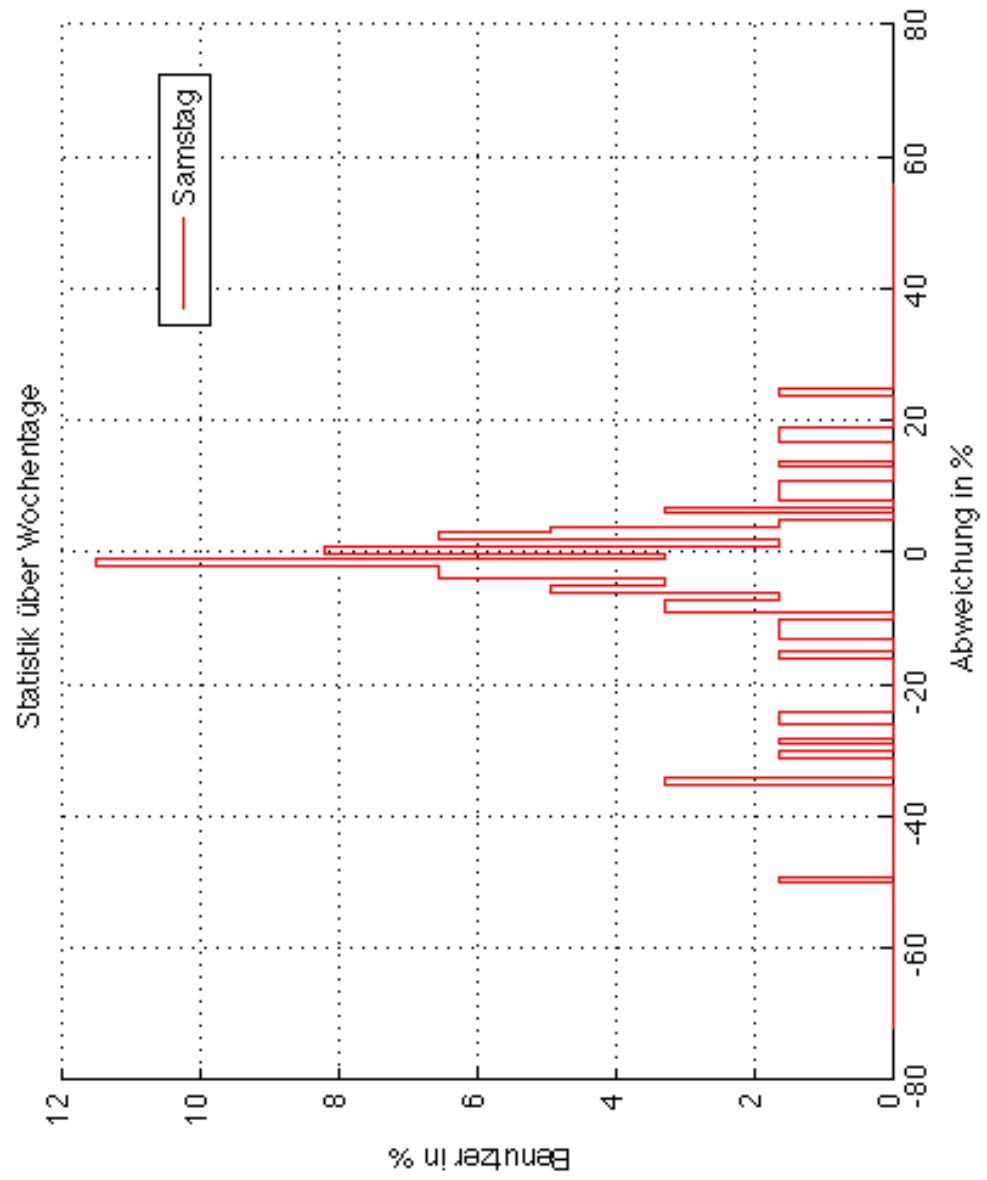


Fig. 13 Auswertung der Samstage über alle Benutzer

## Werktag/Wochenende-Auswertung

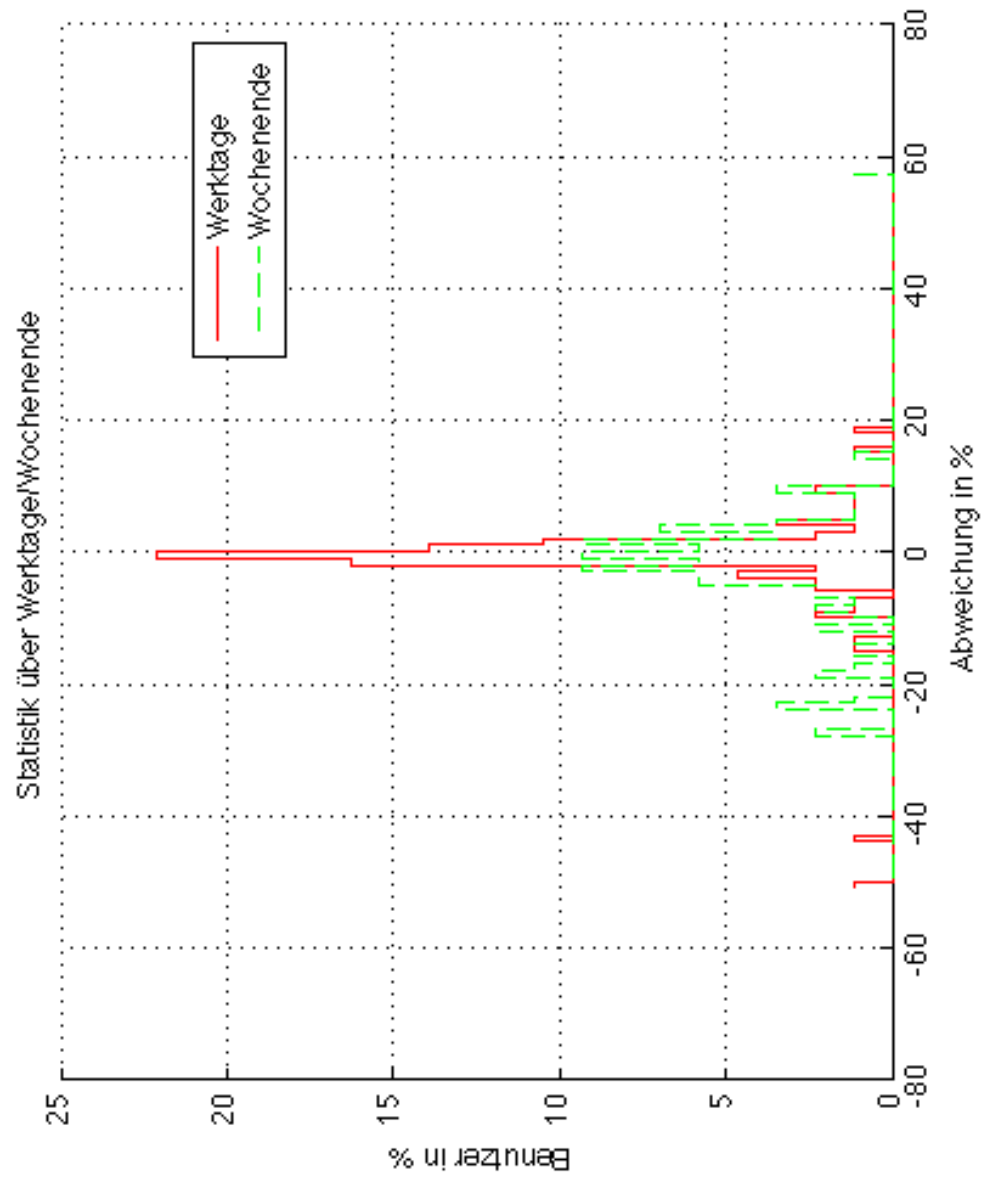


Fig. 14 Auswertung der Werktag/Wochenenden über alle Benutzer

Tageszeit/Wochentag-Auswertung

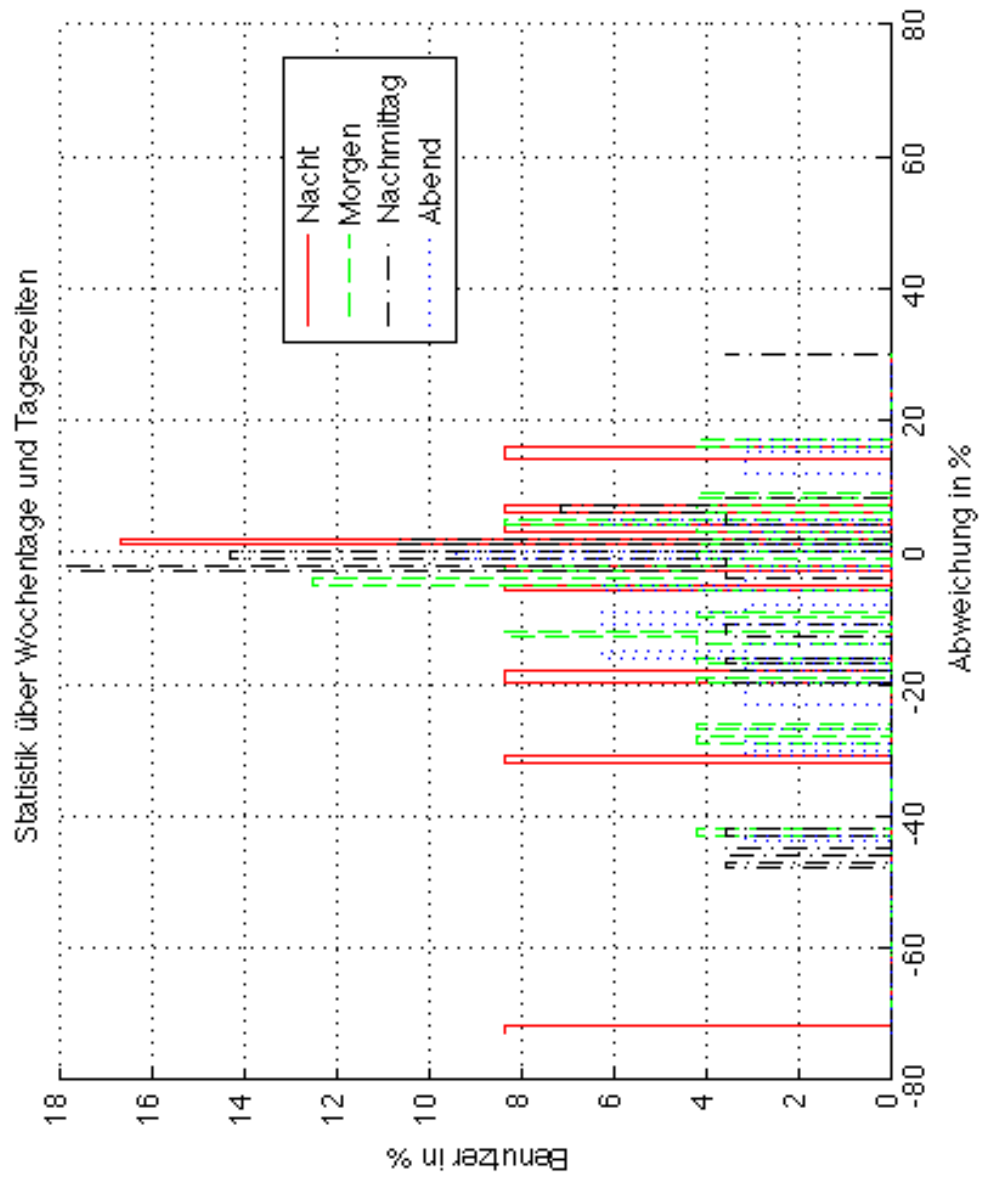


Fig. 15 Auswertung über die Zeit und über den Sonntag über alle Benutzer

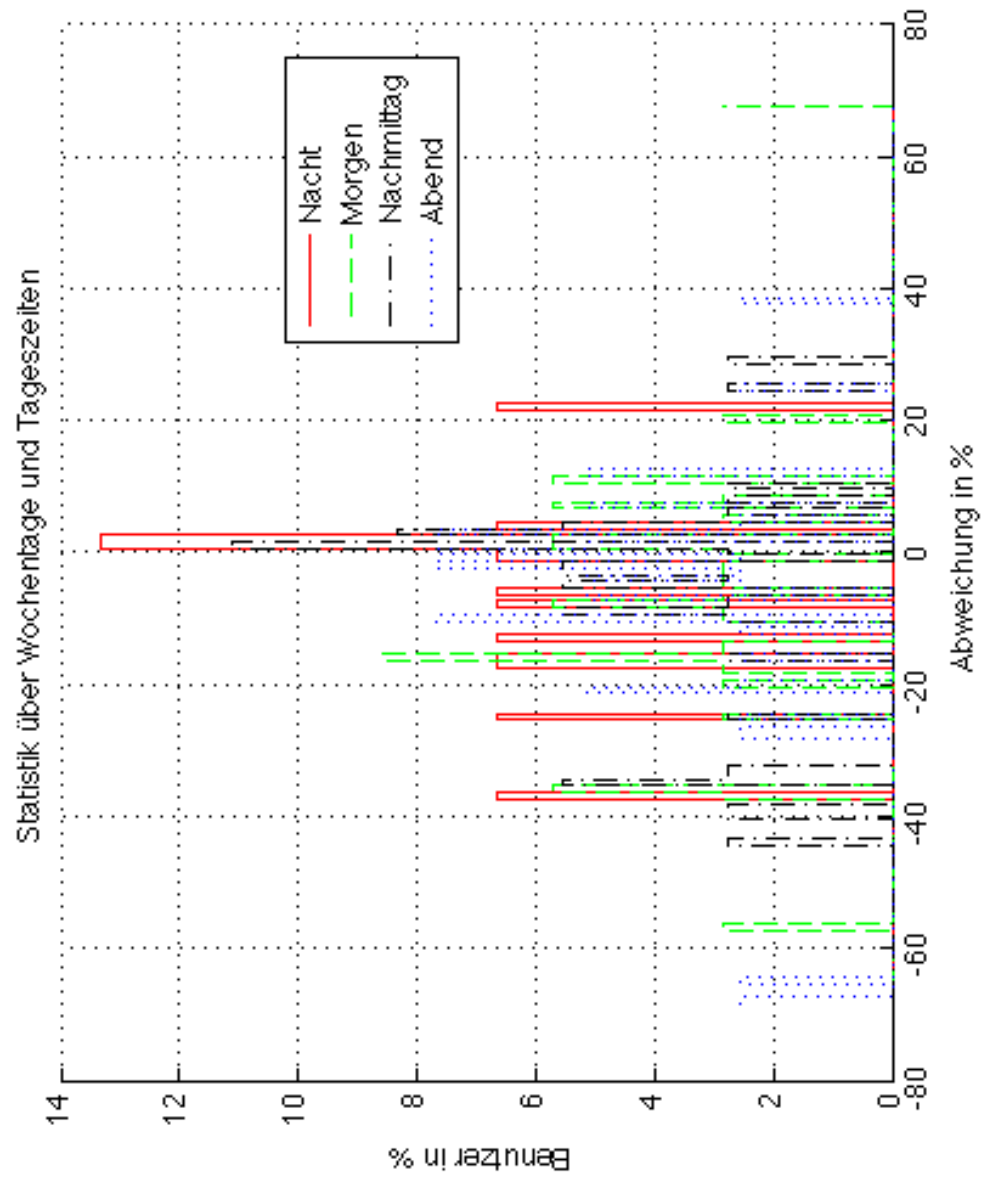


Fig. 16 Auswertung über die Zeit und über den Montag über alle Benutzer

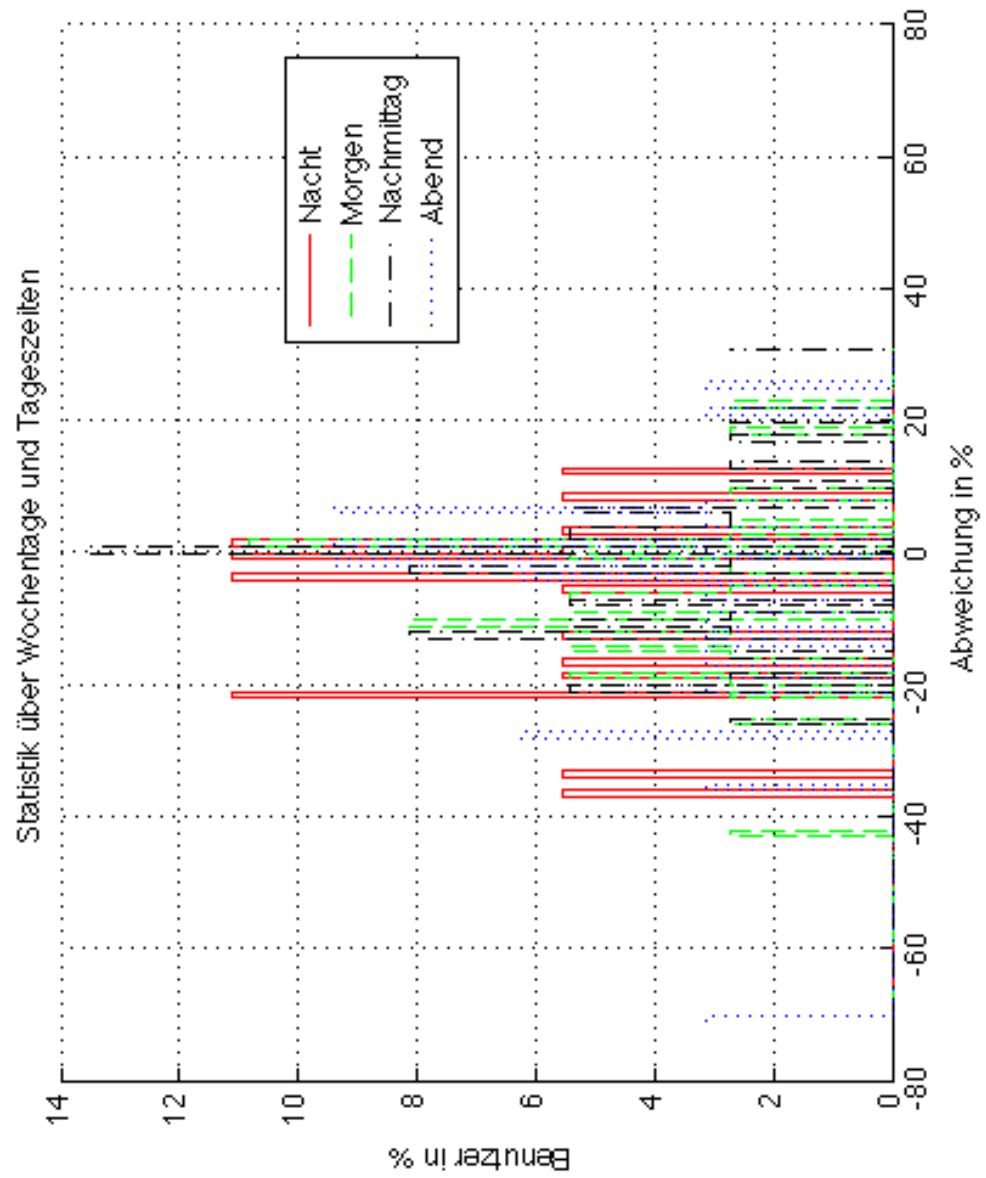


Fig. 17 Auswertung über die Zeit und über den Dienstag über alle Benutzer



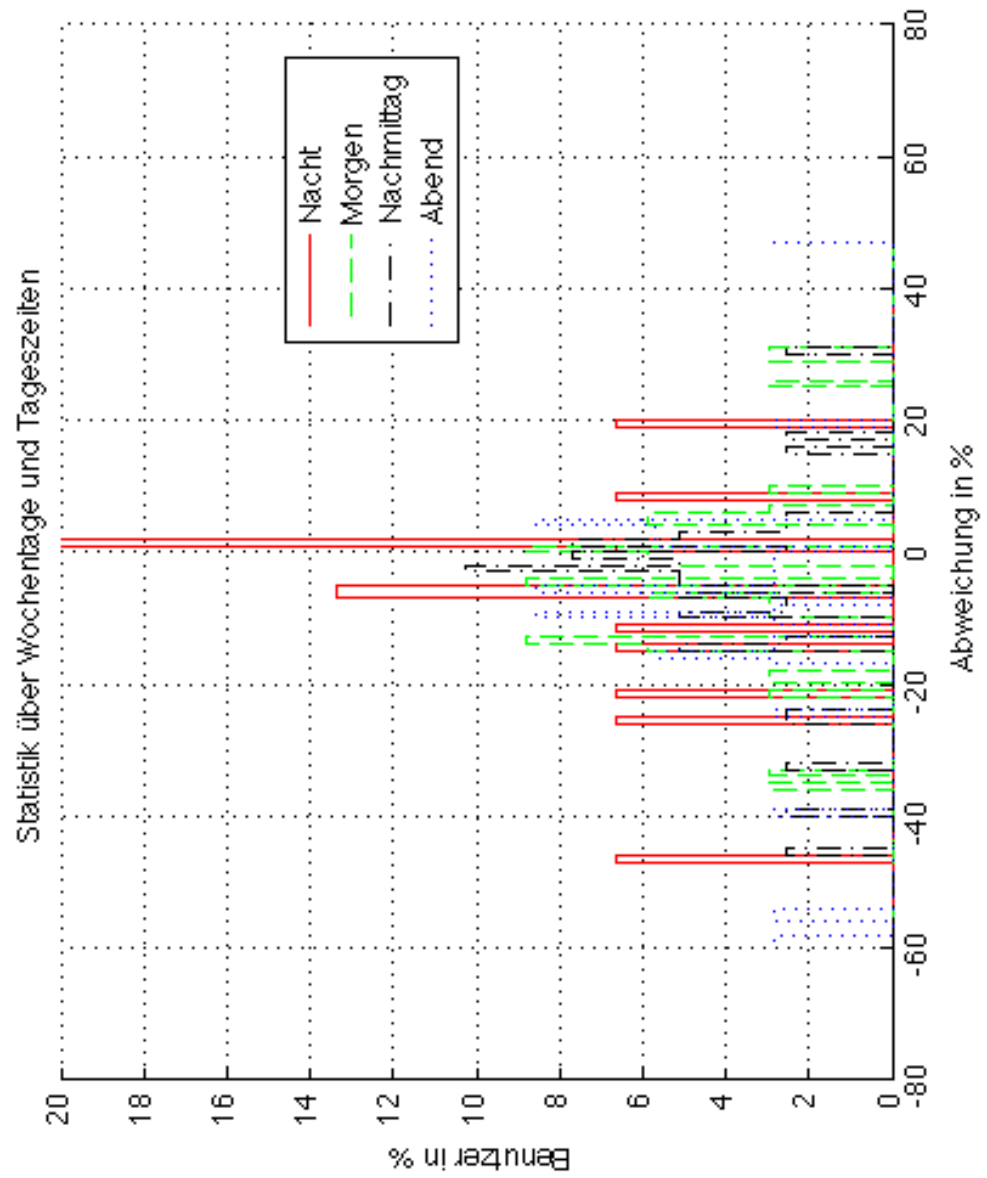


Fig. 18 Auswertung über die Zeit und über den Mittwoch über alle Benutzer

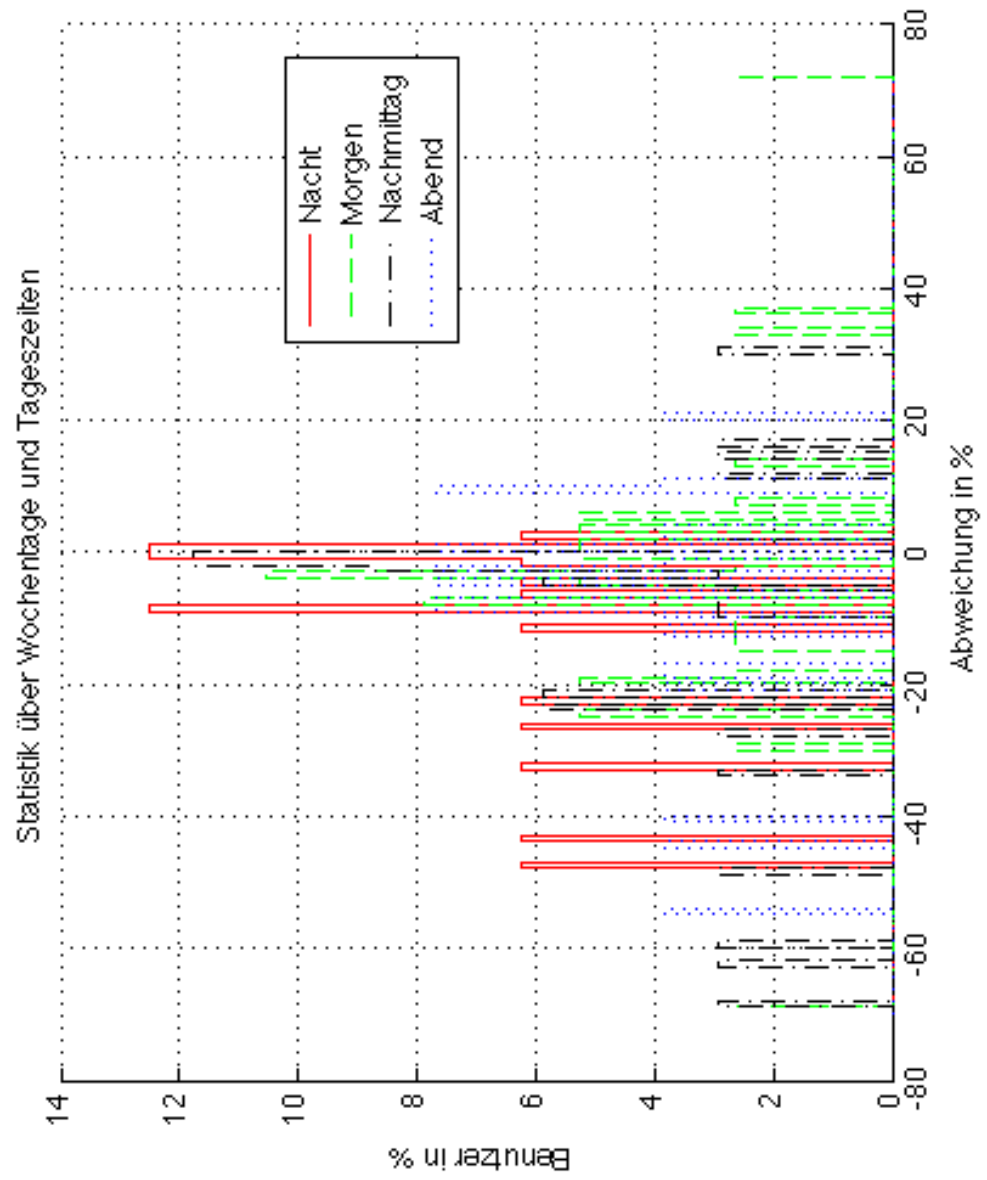


Fig. 19 Auswertung über die Zeit und über den Donnerstag über alle Benutzer

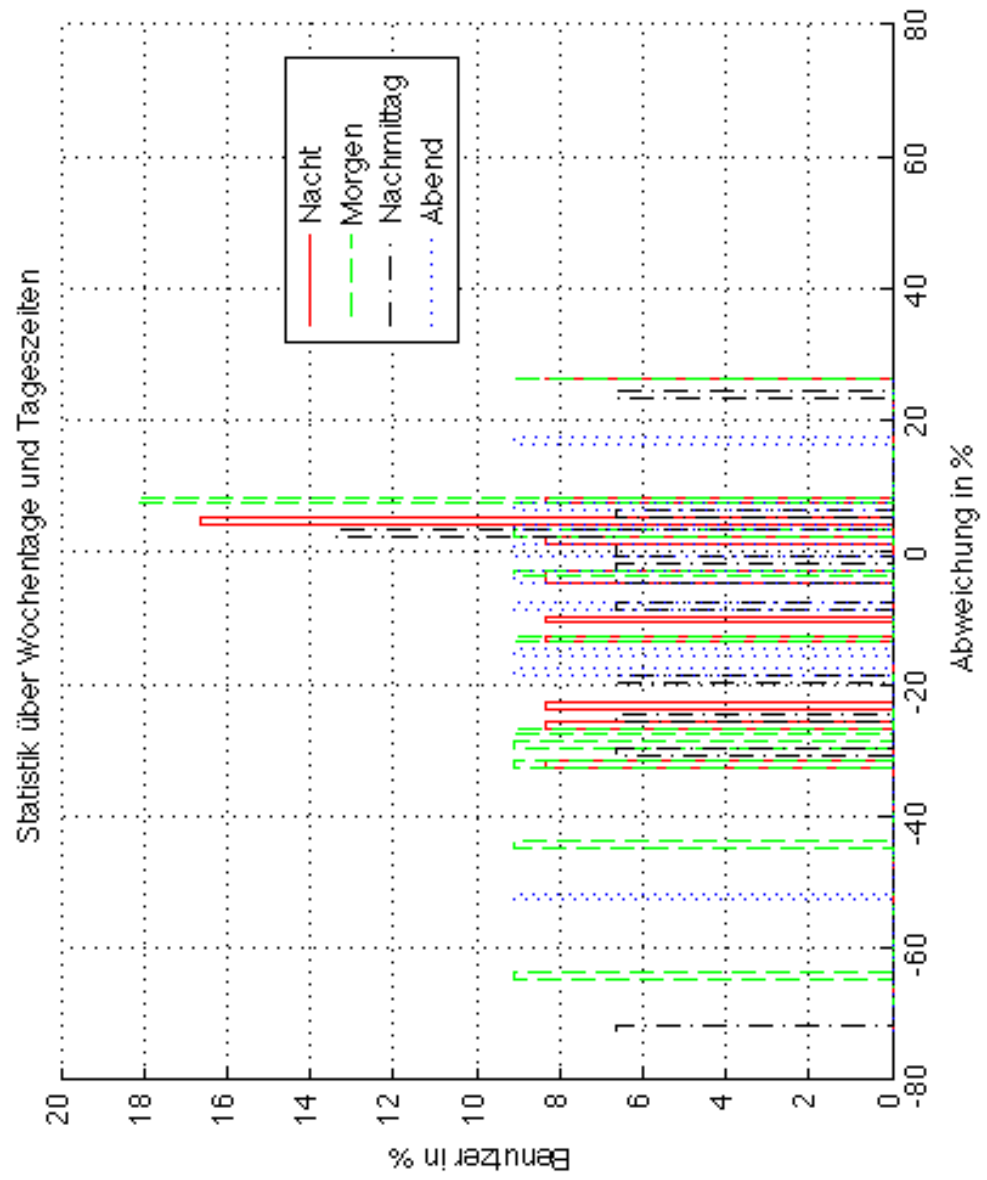


Fig. 20 Auswertung über die Zeit und über den Freitag über alle Benutzer

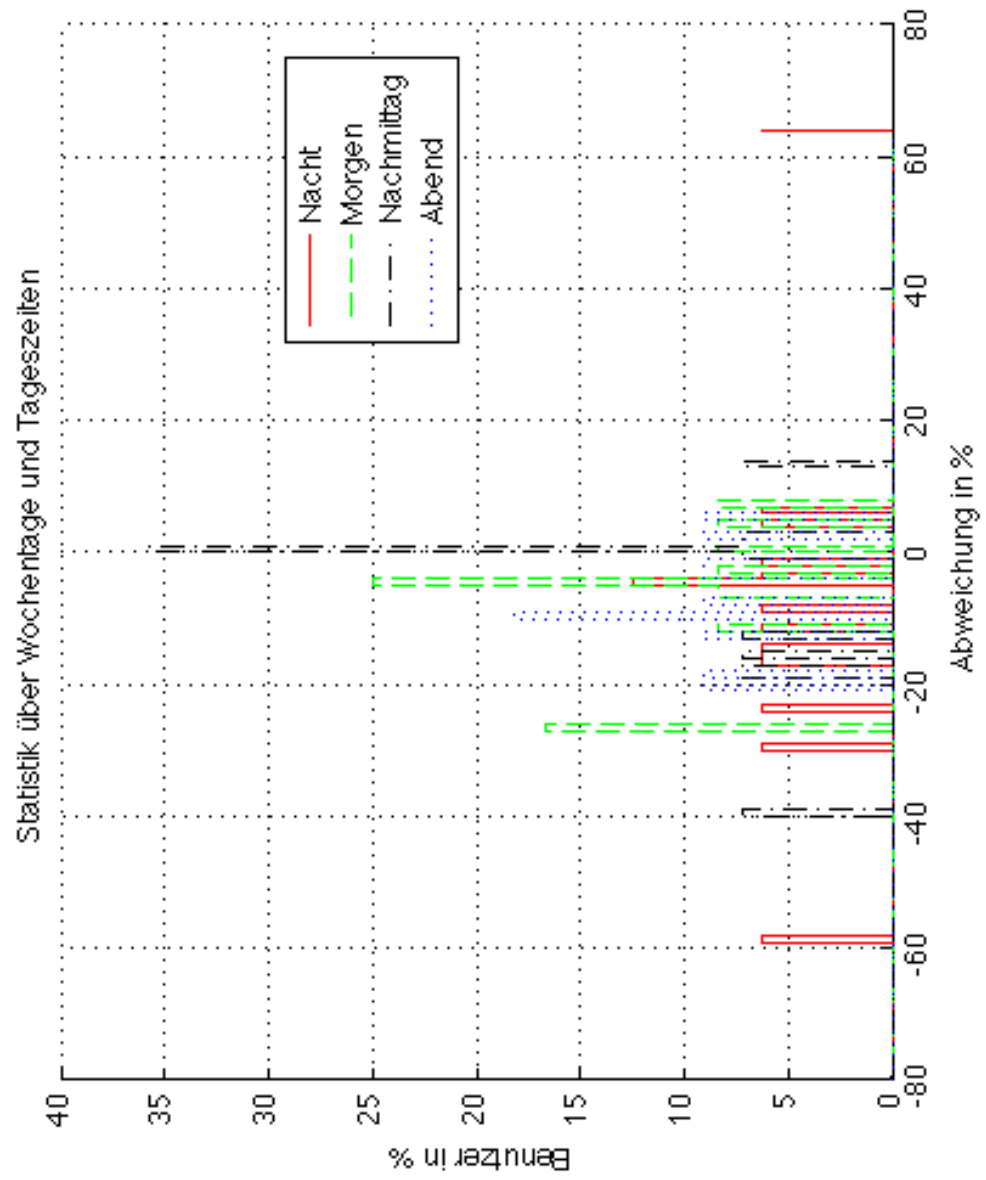


Fig. 21 Auswertung über die Zeit und über den Samstag über alle Benutzer