

Warenkorbanalyse Gruppe B

Code ▾

Vincenzo Timmel, Eric Winter

9. März 2020

Code

Code

Code

Code

Projektauftrag

Wir wurden von Ihnen beauftragt, folgende drei Fragen Mithilfe dieser Daten zu beantworten: + Stärkung und Promotion der Bio-Sparte + eine kunden-freundliche Anordnung der Produkte, so dass Kunden Artikel leichter finden und beim Einkauf schneller zum Ziel kommen (store layout design) + eine Analyse der Ladenöffnungszeiten und Prüfung der Einsatzpläne der Mitarbeiter, welche den Bedürfnissen der Kundschaft und der Angestellten Rechnung trägt

Repräsentativität der Daten

Die Daten sind sehr sauber gehalten und ermöglichen so eine einfache Auswertung. Diese Daten zeigen über einen langen Zeitraum alle Einkäufe, inkl. Zeit und Wochentag des Einkaufes, welche Produkte gekauft wurden, zu welchem Department und Aisle diese gehören und in welcher Reihenfolge diese in den Warenkorb gelegt wurden. Die Daten enthalten also eine grosse Dichte an Informationen und erlauben uns, spezifische, aber auch allgemeingültige Entscheidungen zu treffen.

(Eigentlich gehören diese Daten Instacart, einem Verkaufsservice, welcher einem erlaubt, Leute zu buchen, die für einen den Einkauf erledigen. D.h. diese Daten sind nicht aus einer realen Detailhandel-Verkaufsstelle gesammelt. Diese sind also streng genommen nicht repräsentativ, aber für dieses Projekt sollte man das ja nicht betrachten).

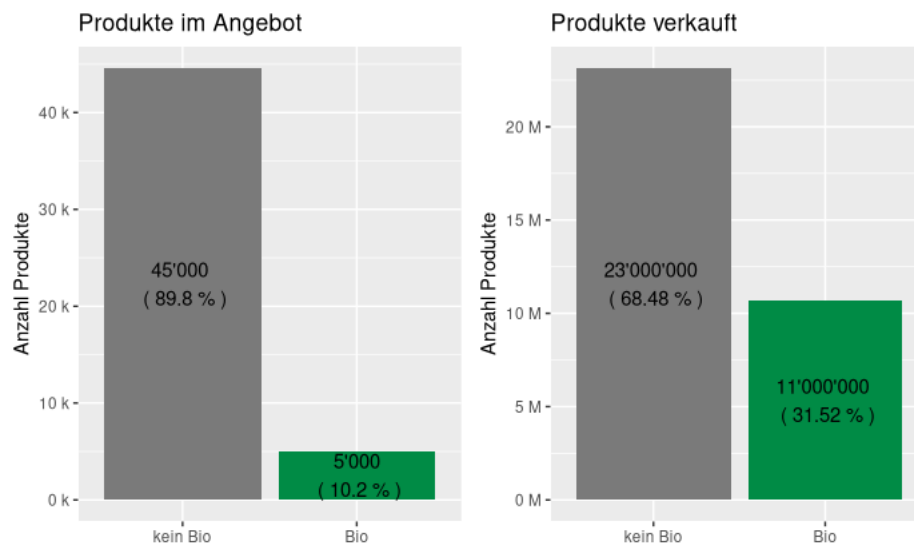
Promotion Bioprodukte

Wie viele Bioprodukte sind im Angebot und wie viele werden tatsächlich verkauft? Gibt es überhaupt Nachfrage für Bioprodukte?

Code

Code

Code



Wenn man die beiden Diagramme mit einander vergleicht, sieht man, dass anteilig mehr Bioprodukte gekauft werden, als sie im Angebot vorhanden sind.

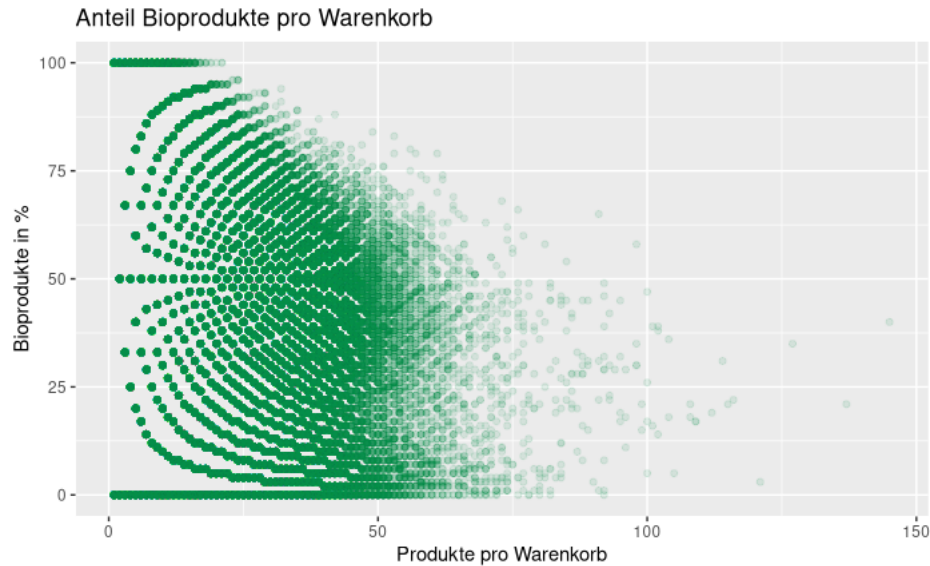
Konkret heisst das:

- Angeboten werden von den insgesamt 49'688 Produkten 5'070 Bioprodukte (etwa 10%)
- Gekauft werden aber von 33'819'106 verschiedenen Produkten 10'659'027 Bioprodukte (etwa 32 %)

Wie sieht der Anteil der Bioprodukte auf Warenkorbebene aus?

	user_id <int>	orders_id <int>	count_orders <dbl>	sum_organic <dbl>	ratio <dbl>
1	112108	1	8	5	0.62
2	202279	2	9	2	0.22
3	205970	3	8	4	0.50

	user_id <int>	orders_id <int>	count_orders <dbl>	sum_organic <dbl>	ratio <dbl>
4	178520	4	13	0	0.00
5	156122	5	26	4	0.15
6	22352	6	3	0	0.00
6 rows					

[Code](#)

Erkennbar ist, dass es durchaus einige Kunden gibt, die gar keine oder ausschliesslich nur Bioprodukte kaufen. Ein weiterer interessanter Punkt ist, dass sich geometrische Muster im Plot bilden. Der Grund hierfür ist, dass es pro Anzahl Produkte im Warenkorb nicht beliebig viele verschiedene Verhältnisse zwischen Bio- und nicht Bioprodukten gibt.

Bei 5 Produkten im Warenkorb gibt es folgende möglichen Kombinationen:

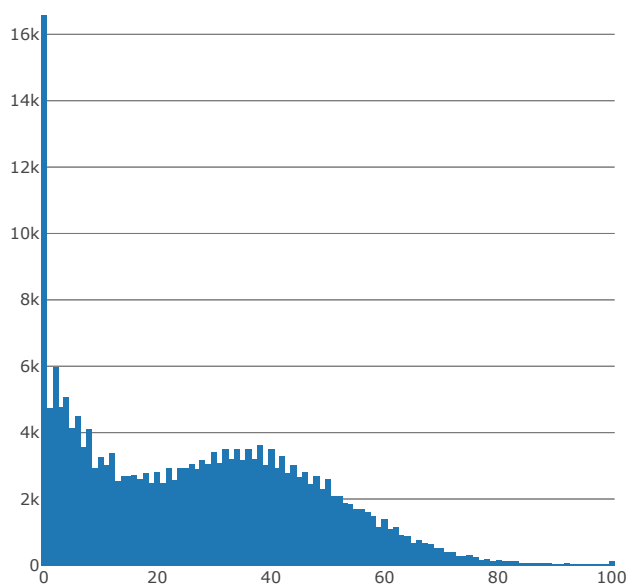
- keine Bioprodukte und 5 nicht Bioprodukte: 0% Bio
- 1 Bioprodukt und 4 nicht Bioprodukte: 20% Bio
- 2 Bioprodukte und 3 nicht Bioprodukte: 40% Bio
- 3 Bioprodukte und 2 nicht Bioprodukte: 60% Bio
- 4 Bioprodukte und 1 nicht Bioprodukt: 80% Bio
- 5 Bioprodukte und keine nicht Bioprodukte: 100% Bio

Somit gibt es an diesen Punkten eine Überlagerung von mehreren Datenpunkten an diesen Stellen und dadurch bilden sich solche Muster.

Je mehr Produkte im Warenkorb liegen, umso mehr Möglichkeiten gibt es.

Bioprodukte auf Ebene Kunde

Prozentualer Anteil Bioprodukte pro Warenkorb



In diesem Plot ist deutlich zu erkennen, dass die viele Kunden unter 50% Anteil Bioprodukte kaufen. Die meisten kaufen sogar gar keine Bioprodukte. Das "Sägezahnmuster" existiert aus den gleichen Gründen zustande wie das Muster in der Grafik "Anteil Bioprodukte pro Warenkorb".

Betrachten wir zusätzlich eine kurze statistische Zusammenfassung der Daten:

Betrachten wir deshalb eine kurze statistische Zusammenfassung der Daten:

[Code](#)

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	8.00	26.00	26.97	42.00	100.00

Hier wird die obige Vermutung bestätigt: Die 75% der Kunden hat in ihren Einkäufen unter 50% Bioprodukte im Warenkorb. Durchschnittlich sind es etwa 27% Bioprodukte pro Warenkorb. Der Maximalwert zeigt uns jedoch, dass es auch Warenkörbe gibt, die ausschließlich aus Bioprodukten bestehen.

Schwankungen beim Kauf von Bioprodukten

Wie stark schwankt der Anteil Bioprodukte bei Kunden?

[Code](#)

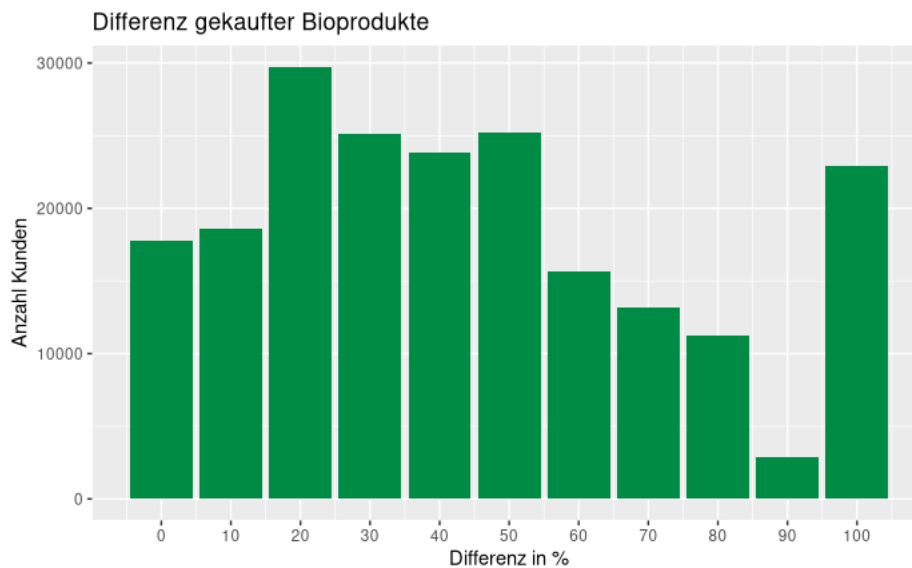
Auf der X-Achse ist die Differenz des maximalen und minimalen prozentualen Anteils der Kunden zusehen. Die Werte sind auf die nächsten 10% gerundet.

Beispiel: Eine Kundin kauft mit folgenden Anteilen Bioprodukte ein:

- 54%
- 35%
- 78%
- 61%

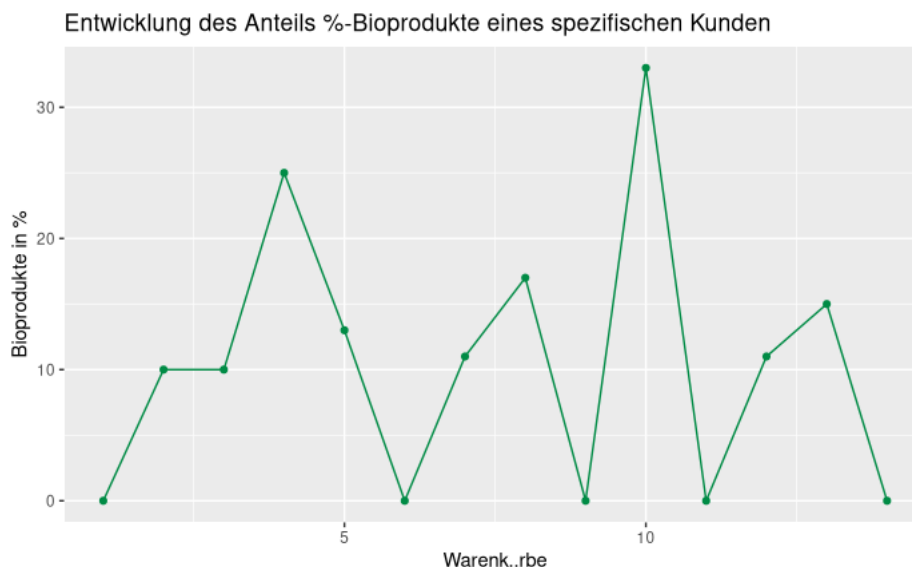
Die Differenz des minimalen und maximalen Wertes sind $78\% - 35\% = 43\%$. Dieses Ergebnis wird auf 40% abgerundet.

Betrachten wir diese Differenz für jeden Kunden, gibt sie uns Auskunft darüber, wie stark das Einkaufsverhalten der Kunden schwankt. Kunden die sowieso schon Bioprodukte kaufen, müssen nicht mehr in Promotionsaktionen mit eingeschlossen werden und man kann dort Kosten sparen. Auf der anderen Seite sind Leute die eine hohe Differenz des Bioanteils bei ihren Einkäufen zeigen, eher gute Ziele für Promotionsaktionen.



Bioanteil einzelner Kunden betrachten (Microtargeting)

Betrachtet man das Einkaufsverhalten eines einzelnen Kunden im Bezug auf gekaufte Bioprodukte, können die bereits vermuteten Schwankungen isoliert werden



Da dieser Kunde immer einen schwankenden Anteil Bioprodukte kauft, wäre er gut geeignet als Ziel für Promotionsaktionen, da er bereit ist Bioprodukte zu kaufen, das aber nur unter gewissen Umständen. Wenn solche Kunden einen Grund haben, können sie dauerhafte “Biokunden” werden. Beispielsweise durch gezielt zugestellte Gutscheine, Aufklärungskampagnen oder Ähnliches.

Assoziationanalyse

Code

Joining, by = "product_id"Joining, by = "orders_id"Joining, by = "aisle_id"

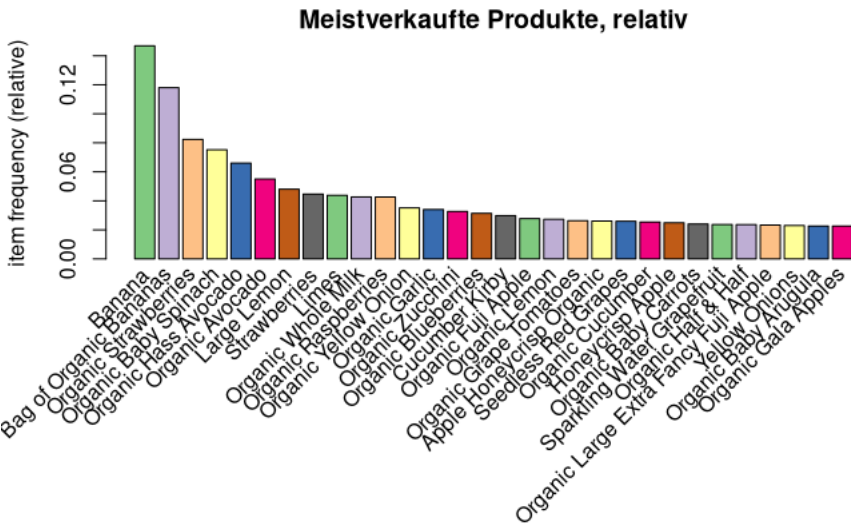
Code

Code

Code

Mit Hilfe einer Assoziationanalyse kann man herausfinden, welche Produkte in Kombination mit einander gekauft werden. So kann man Kunden, die 3 von 4 gängigen Produkten kaufen, das vierte empfehlen und die Wahrscheinlichkeit, dass es tatsächlich gekauft wird ist hoch.

Auch kann man sehen, welche Produkte wie häufig gekauft werden. Im folgenden Plot sieht man beispielsweise, dass Früchte sehr häufig gekauft werden. Auf der Y-Achse kann man den prozentualen Anteil der Häufigkeit ablesen.



Code

Apriori

Parameter specification:

confidence	minval	smax	arem	aval	originalSupport	maxtime	support	minlen
<dbl>	<dbl>	<dbl>	<fctr>	<lgl>	<lgl>	<dbl>	<dbl>	<int>
0.3	0.1	1	none	FALSE	TRUE	5	0.001	2

1 row | 1-10 of 12 columns

Algorithmic control:

filter	tree	heap	memopt	load	sort	verbose
<dbl>	<lgl>	<lgl>	<lgl>	<lgl>	<int>	<lgl>
0.1	TRUE	TRUE	FALSE	TRUE	2	TRUE

1 row

Absolute minimum support count: 3346

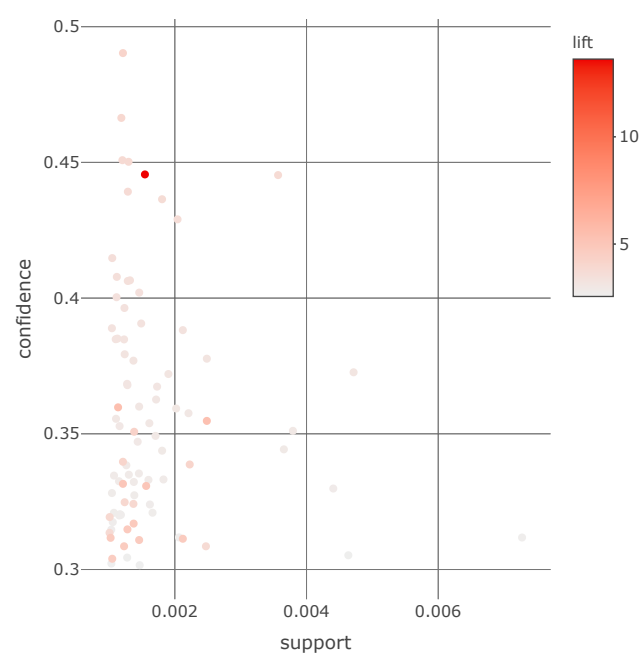
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[49701 item(s), 3346084 transaction(s)] done [14.27s].
sorting and recoding items ... [1776 item(s)] done [0.32s].
creating transaction tree ... done [2.88s].
checking subsets of size 1 2 3 4 done [0.87s].
writing ... [174 rule(s)] done [0.01s].
creating S4 object ... done [1.11s].

Code

Cross-Selling für organische Produkte

Cross-Selling beschreibt das zusätzliche Verkaufen von Dienstleistungen oder Produkten. Ein Vorgehen, um Crossselling zu erfolgreich umzusetzen, ist der Apriori-Algorithmus. Dieser Algorithmus analysiert Warenkörbe und zeigt auf, welche Produkte oft zusammen gekauft werden genannt Regel, und wie oft diese Regel vorkommt. Diese Angabe wird vom Algorithmus zurückgegeben als “Support”, “Confidence” und “Lift”. Der Support zeigt uns die relative Häufigkeit dieser Regel, also wie oft diese Produkte zusammen in einem Warenkorb laden, geteilt durch die Anzahl von allen Warenkörbe. Der Confidence ist wie oft zwei Produkte zusammengekauft werden, geteilt durch das Produkt, auf dem die Regel basieren soll. Der Lift sagt an, wie stark diese Regel von der Norm abweicht und ob diese Produkte positiv, negativ oder gar nicht miteinander korrelieren. Der Lift entsteht, wenn man den Support durch das Vorkommen von beiden Produkten teilt.

Lesebeispiel für den tiefroten Punkt: Wenn man {Organic Yellow Squash} kauft, kauft man mit 44.6% Wahrscheinlichkeit auch {Organic Zucchini}. Falls also ein Kunde nur {Organic Yellow Squash} kauft, kann man diesem Kunden einen persönliche Gutschein für {Organic Zucchini} zustellen oder im Allgemeinen diese nahe beieinander positionieren. Der Support sagt uns, dass diese Produkte in 0.0000155% der Warenkörbe zusammen vorkommen und der Lift, dass diese Regel stark von der Norm abweicht und diese Produkte deshalb stark zusammen korrelieren.



Code

```
Joining, by = "orders_id"Joining, by = "product_id"
```

Code

Code

Layout

Statistik über Anzahl Produkte in den Warenkörben

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.00	5.00	8.00	10.11	14.00	145.00

Bestehendes Layout

Hier sehen wir das ursprüngliche Layout. Es unterscheidet sich von dem Layout, das wir erhalten haben. Um das bestehende Layout zu finden, kreierten wir eine Liste mit allen Departments. Je früher und je häufiger ein Artikel aus einem Department (via add_to_cart_order) in den Warenkorb gelegt wird, desto früher taucht dieses in der Liste auf. “Dairy eggs” ist beispielsweise sehr oft als zweites in den Warenkörben gelandet und nicht erst gegen Ende wie man es im ursprünglichen Layout erwarten würde. Beiliegend zu diesem Skript findet man ein auf dieser Liste basierendes PDF mit diesem generierten Layout.

Code

Code

department <chr>	min_add <dbl>	products_bought_sum <dbl>
produce	1	4218888
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
dairy eggs	2	1525274
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
snacks	3	557341
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
frozen	4	356535
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
pantry	5	242257
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
beverages	6	133972
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
canned goods	7	89909
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
dry goods pasta	8	58810
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
deli	9	25476
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
household	10	14744
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
personal care	11	6808
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
breakfast	12	2801
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
meat seafood	13	996
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
international	14	393
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
---------------------	------------------	------------------------------

department <chr>	min_add <dbl>	products_bought_sum <dbl>
bakery	15	173
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
babies	16	87
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
missing	17	42
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
alcohol	18	21
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
other	19	20
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
pets	20	22
1 row		

department <chr>	min_add <dbl>	products_bought_sum <dbl>
bulk	21	7
1 row		

Code

```
$department
[1] "produce"

$department
[1] "dairy eggs"

$department
[1] "snacks"

$department
[1] "frozen"

$department
[1] "pantry"

$department
[1] "beverages"

$department
[1] "canned goods"

$department
[1] "dry goods pasta"

$department
[1] "deli"

$department
[1] "household"

$department
[1] "personal care"

$department
[1] "breakfast"

$department
[1] "meat seafood"

$department
[1] "international"

$department
[1] "bakery"

$department
[1] "babies"

$department
[1] "missing"

$department
[1] "alcohol"

$department
[1] "other"

$department
[1] "pets"

$department
[1] "bulk"
```

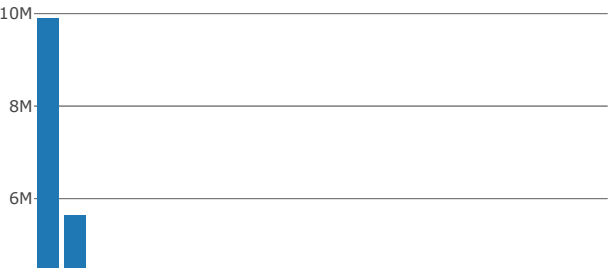
Optimiertes Layout

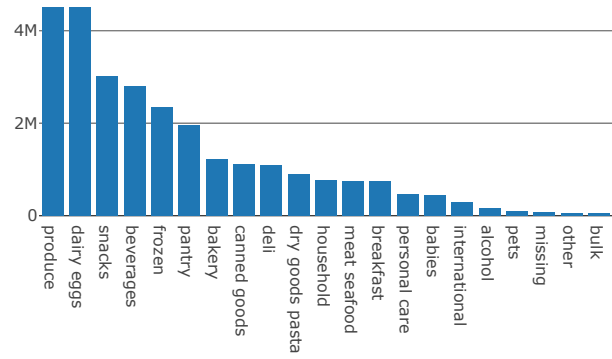
Im Plot sind die Departments nach Verkaufszahlen angeordnet. Da es viele diverse Warenkörbe gibt, setzt sich das optimale Layout anhand der Verkaufszahlen der einzelnen Departments zusammen. Departments mit hohen Verkaufszahlen befinden sich eher am Anfang. Ebenfalls hat es ein Department für Aisles, welche in kleinen Warenkörben vorkommen (Folgt gleich nach dieser Information). Beiliegend zu diesem Skript findet man ein auf dieser Liste basierendes PDF, welches das optimierte Layout anzeigt.

Code

Code

Von welchen Departments wird am meisten bestellt?



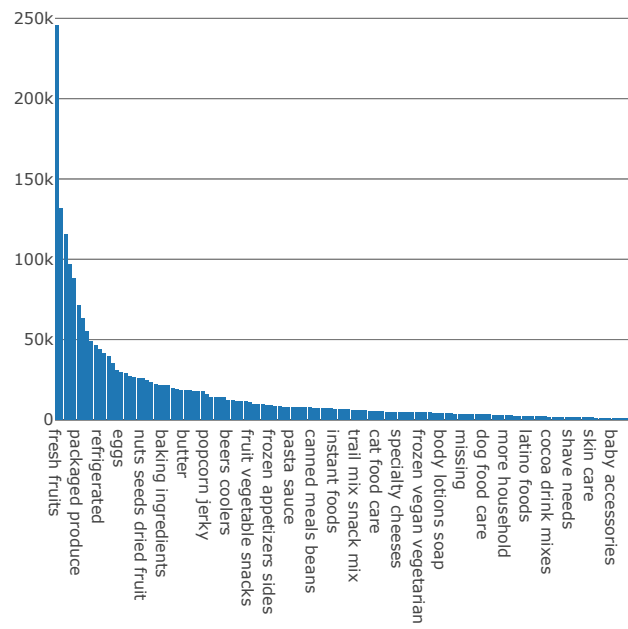


Welche Aisles kommen häufig in kleinen Warenkörben vor?

Betrachtet man nur Warenkörbe mit weniger als vier Produkten, kann man zusätzliche Optimierungen im Layout vornehmen. Die Aisles, die oft in diesen Warenkörben vorkommen, sollte man nahe am Eingang / Ausgang stellen, um Leuten einen schnellen und kleinen Einkauf zu ermöglichen.

Code
Code

Hufige Aisles in kleinen Warenkörben



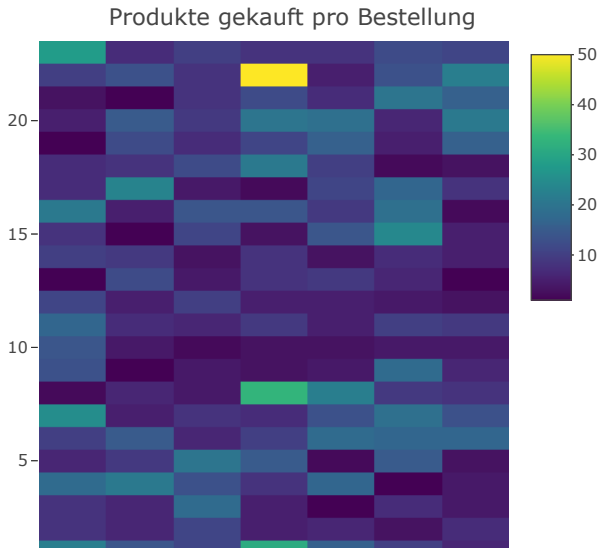
Warenkorb-Grösse nach Zeit und Tag

Man sieht auch sehr gut, dass in der Mittagszeit die Anzahl Produkte pro Warenkorb abnimmt. Das sind kleine Warenkörbe mit einem Mittagessen (e.g. Frucht, yogurt und ein Sandwich) drin. Der "Mittagessen"-Rush könnte gut von Self-Checkout Maschinen abgefangen werden.

Code

```
Error in eval(lhs, parent, parent) :  
  object 'all_useful_data_joined_db' not found
```

Code





Analyse Öffnungszeiten

Code

```
Joining, by = "order_dow"
```

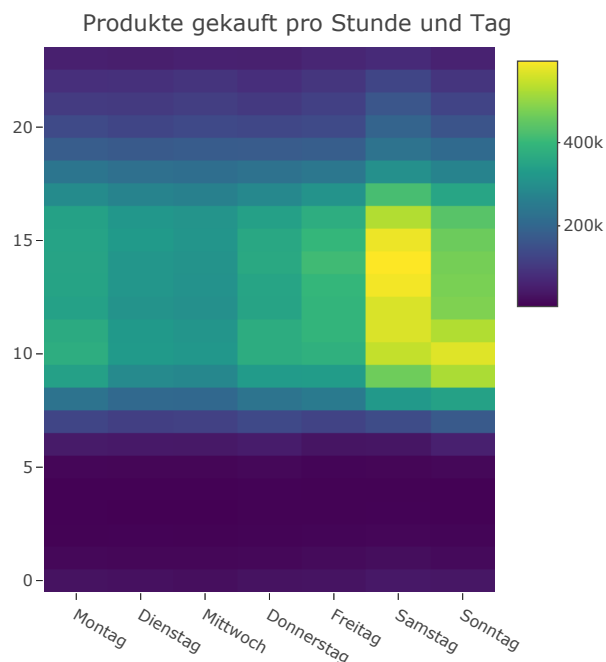
Code

Verkaufte Produkte pro Stunde und Tag

In diesem Plot ist die y-Achse die Uhrzeit und die Farben die Verkaufte Produkte pro Stunde pro Tag. Schaut man sich die gekauften Produkte pro Stunde über alle Wochentage an, ist erkennbar, dass am Wochenende am meisten Produkte gekauft werden. Man kann auch für jede einzelne Stunde nachschauen, wie viele Produkte verkauft werden.

Beispielsweise: Am Samstag von 14-15 Uhr wurden 595'896 Produkte verkauft.

Code



Code

```
Joining, by = "order_dow"
```

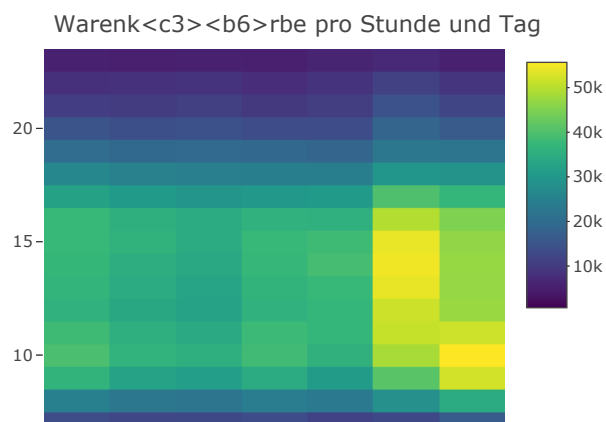
Code

Verkaufte Warenkörbe pro Stunde

Hier schauen wir uns die Bestellungen pro Stunde an. Man kann erkennen, dass am Wochenende die meisten Einkäufe getätigt werden. Auch wird Samstags noch später eingekauft als an den anderen Tagen. Samstags finden die meisten Einkäufe zwischen 10 und 16 Uhr statt, Sonntags 9 und 11 Uhr. Am Wochenende muss man also mit mehr Ansturm rechnen.

Beispiel: Am Samstag von 14-15 Uhr haben wir 54'552 Bestellungen erhalten.

Code





Code

```
Joining, by = "order_dow"
```

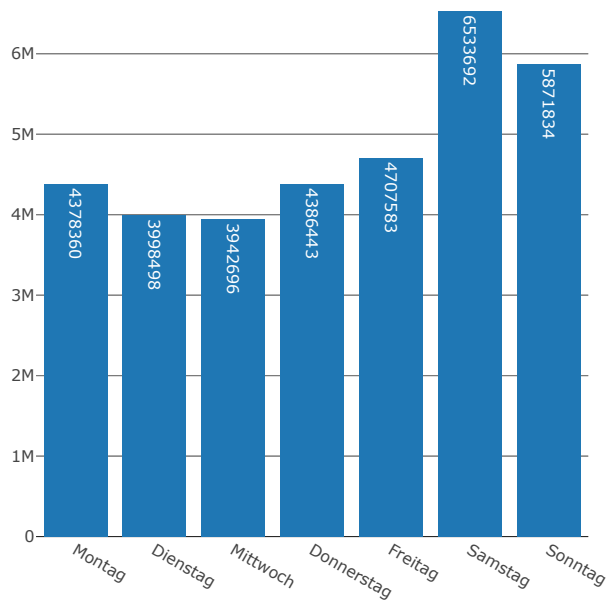
Code

Verkaufte Produkte pro Tag

Im Säulendiagramm sieht man die bereits erkannten Muster. Das Wochenende hat die umsatzstärksten Tage, Dienstag und Mittwoch sind die beiden schwächsten Tage. Im Durchschnitt wird an einem Samstag ca. 1.66x so viele Produkte verkauft wie an einem Mittwoch.

Code

Produkte gekauft pro Tag



Häufigkeit der Tage im Datenset

Man sieht, dass der Samstag häufiger vorkommt im Datenset, unabhängig von den gekauften Produkten. Dies ist aber auch zu erwarten, es bedeutet also nicht, dass am Samstag zuverlässiger Daten erfasst wurden. Am Wochenende wird einfach mehr eingekauft. Der Samstag kommt also dementsprechend 1.41x soviel mal vor wie der Mittwoch.

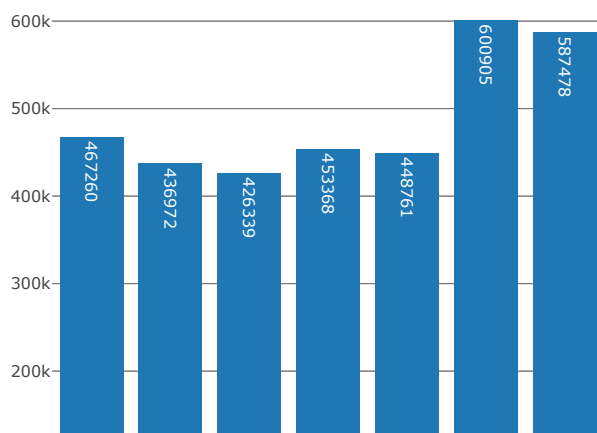
Code

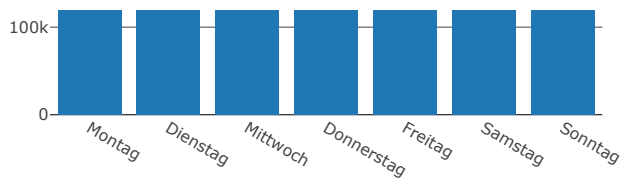
```
Joining, by = "order_dow"
```

Code

Code

Verteilung der Tage im Datenset





Verhältniss der gekauften Produkte pro Tag und Verteilung der Tage im Datenset (Durchschnitte Warenkorbgrösse pro Tag)

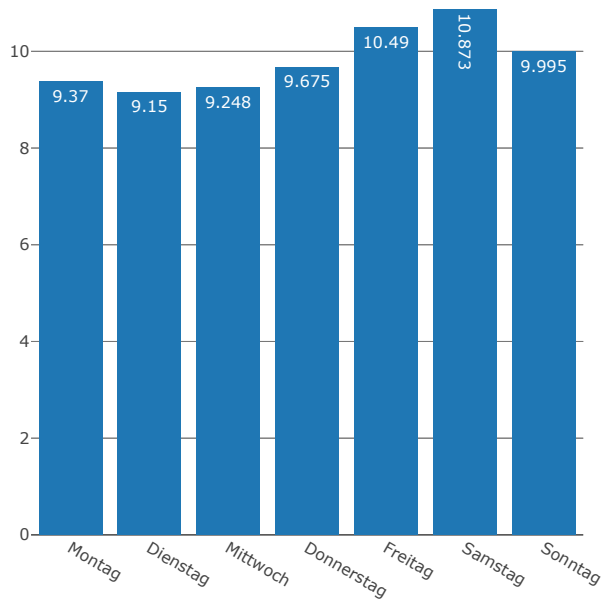
Wir sehen, dass bei der Division von den verkauften Produkten pro Tag durch die Häufigkeit der Tage, das Wochenende ebenfalls auffällt. Am Wochenende wird viel eingekauft, gleichzeitig sind aber auch Warenkörbe am grössten.

Code

```
Joining, by = "order_dow"
```

Code

Durchschnitte Warenkorbgrösse pro Tag



Verkaufte Produkte pro Stunde

Code

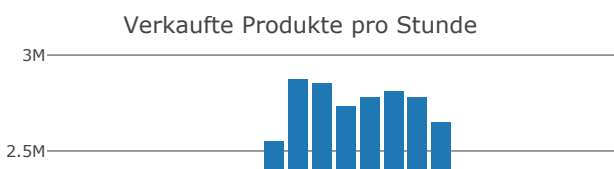
	order_hour_of_day <int>	products_bought <dbl>
1	0	228031
2	1	121412
3	2	72660
4	3	53759
5	4	55714
6	5	91909
7	6	302642
8	7	928239
9	8	1787359
10	9	2550569

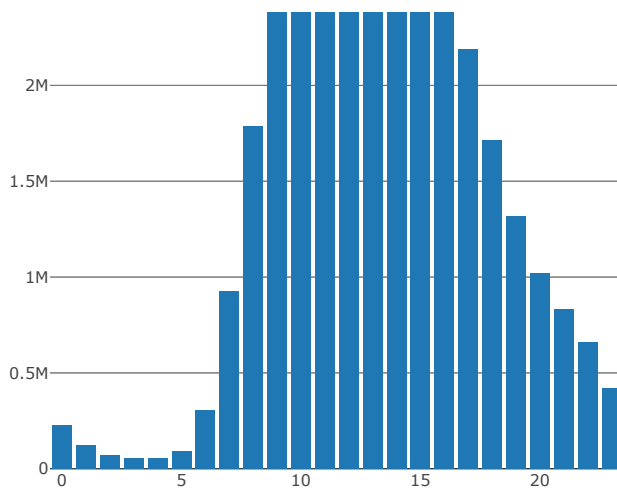
1-10 of 24 rows

Previous 1 2 3 Next

Betrachtet man die gekauften Produkte und zu welcher Uhrzeit sie gekauft wurden, kann man erkennen dass zu den üblichen Öffnungszeiten am meisten Produkte verkauft werden. Der absolute Peak ist aber von 10 Uhr bis 15 Uhr.

Code





Gekaufte Produkte pro Stunde - einzelne Tage im Vergleich

Hier kann man direkt die einzelnen Tage miteinander vergleichen. Interessant ist der Knick am Sonntag. Wahrscheinlich tätigen da viele Leute ihren Morgeneinkauf. Insgesamt ist das Muster aber über alle Tage das gleiche: Zwischen 10 und 16 Uhr wird am meisten eingekauft und nachts so gut wie gar nicht.

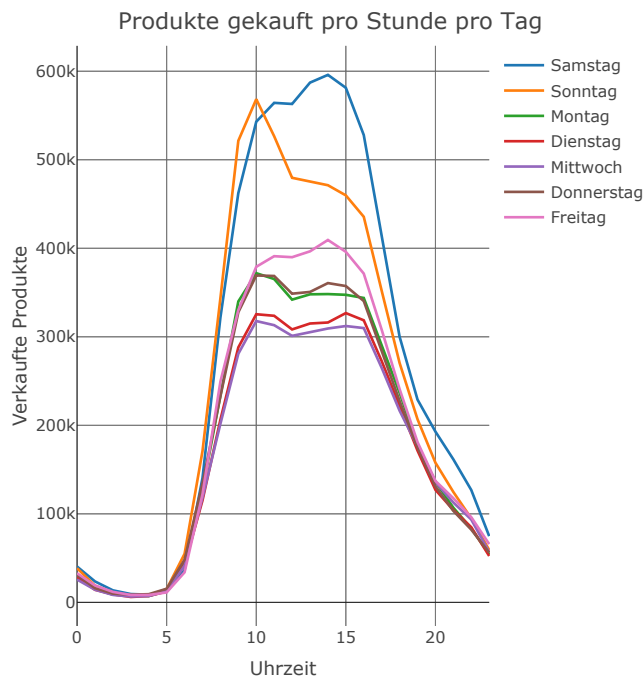
Code

Code

Code

```
Joining, by = "order_dow"
```

Code



Code

Code

Verkaufte Produkte pro Zeit pro Tag (Relativ)

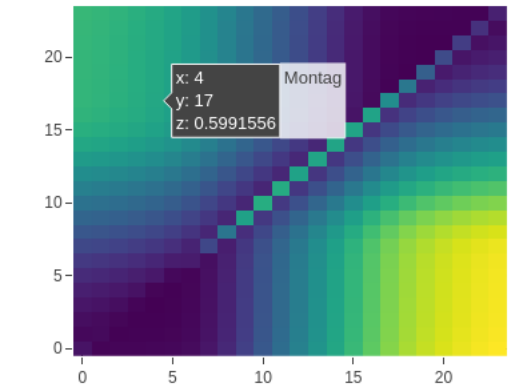
Code

Code

Code

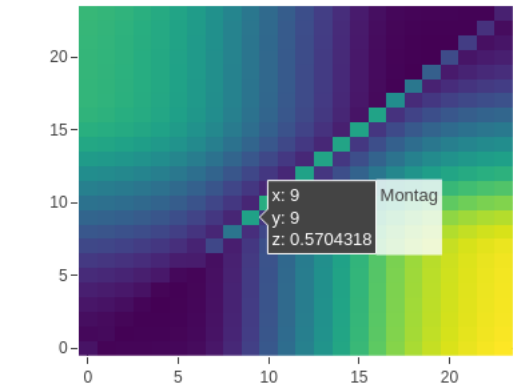
Interaktiver Plot mit den verkauften Produkte pro Stunde relativ zu zwei Referenzen und stündliche erwartete Belastung

Jede der sieben Matrizen zeigt einen Wochentag an und jede einzelne Matrix enthält drei verschiedene Informationen; Oben links der Matrix werden die relativ zum Samstag stündlich verkauften Produkte dargestellt. Lesebeispiel: Am Montag zwischen 4 und 17 Uhr wurden 60.0% der am Samstag verkauften Produkte verkauft.

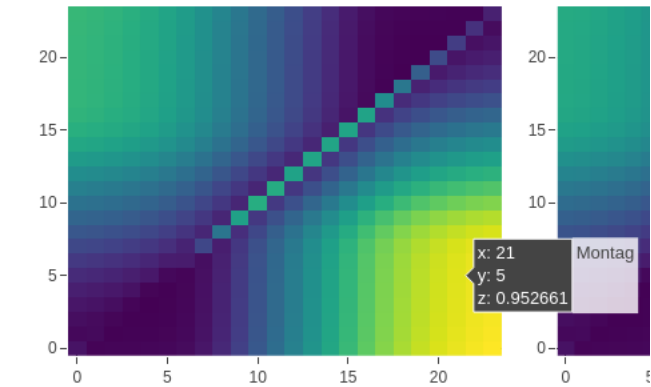


Relativ-verkaufte Produkte im Zeitfenster

In der Diagonalen kann die erwartete Belastung, verglichen mit der Stunde am Samstag zwischen 14-15 Uhr, abgelesen werden. Lesebeispiel:
Am Montag von 9-10 Uhr erwarten wir, dass 57.0% der Produkte wie am Samstag von 14-15 Uhr verkauft werden.



Unten Rechts in der Matrix sehen wir das gleiche wie Oben Links, nur das der Referenz der gleiche Tag ist und nicht der Samstag. Lesebeispiel:
Zwischen 5-21 Uhr am Montag 95.3% der Produkte vom gesamten Tag verkauft.



Aus diesen Angaben lassen sich Arbeitspläne und Einsätze generieren, da man ablesen kann, was man für eine Belastung in einer bestimmten Stunde erwarten kann. Ebenfalls lassen sich damit schnelle und fundierte Entscheidungen treffen bei Ausfällen oder Einsätzen.

Code

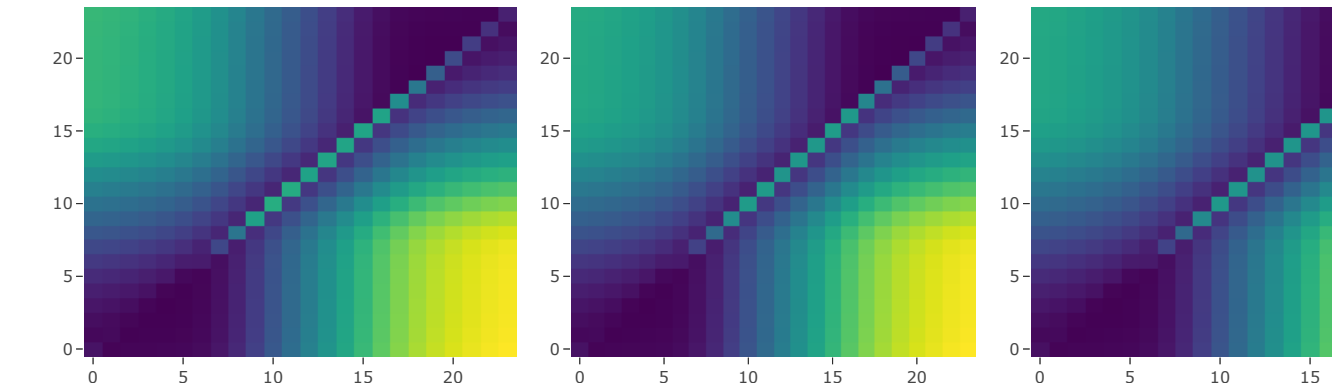
Code

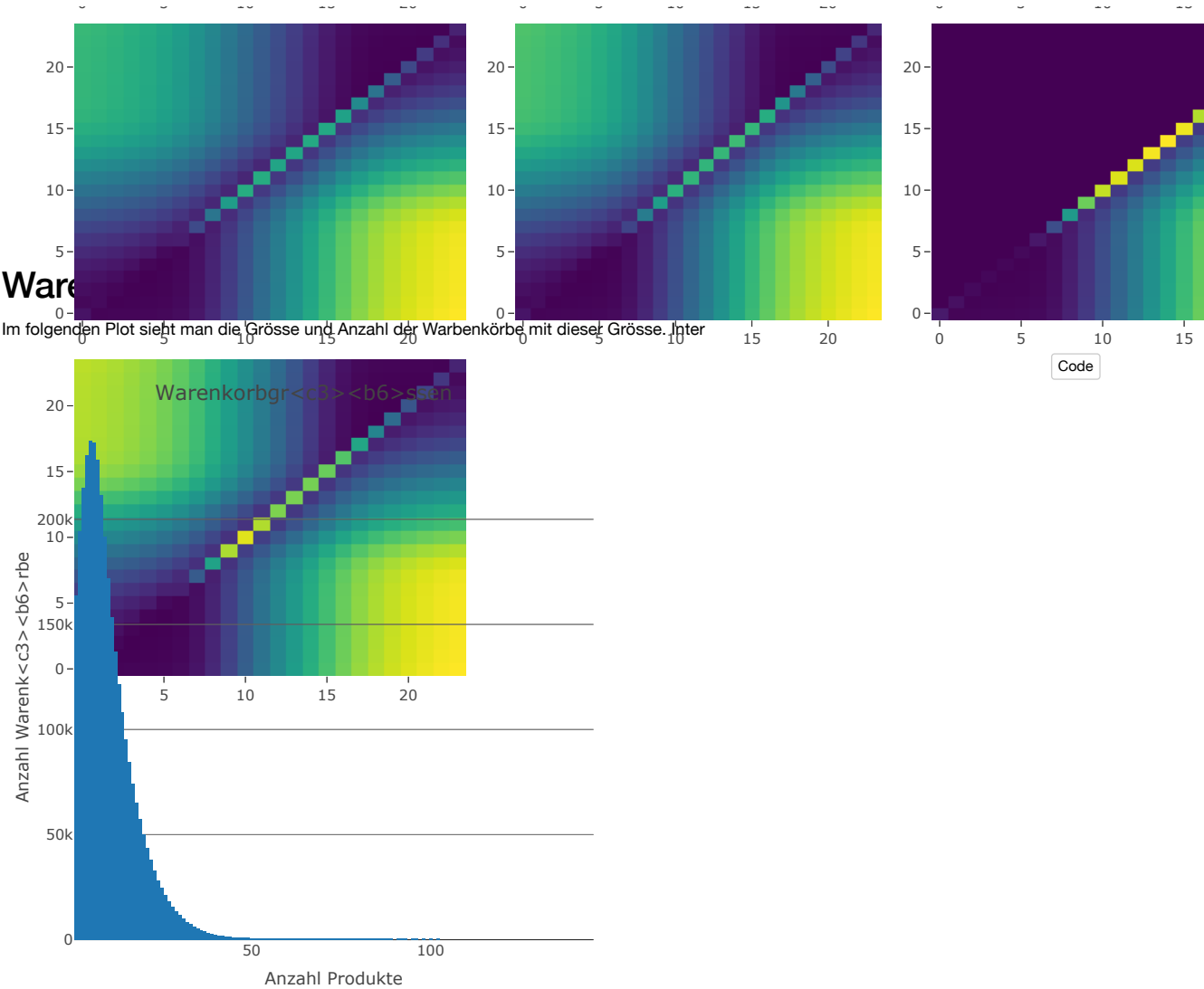
Code

Code

Specifying width/height in layout() is now deprecated.
Please specify in ggplotly() or plot_ly()

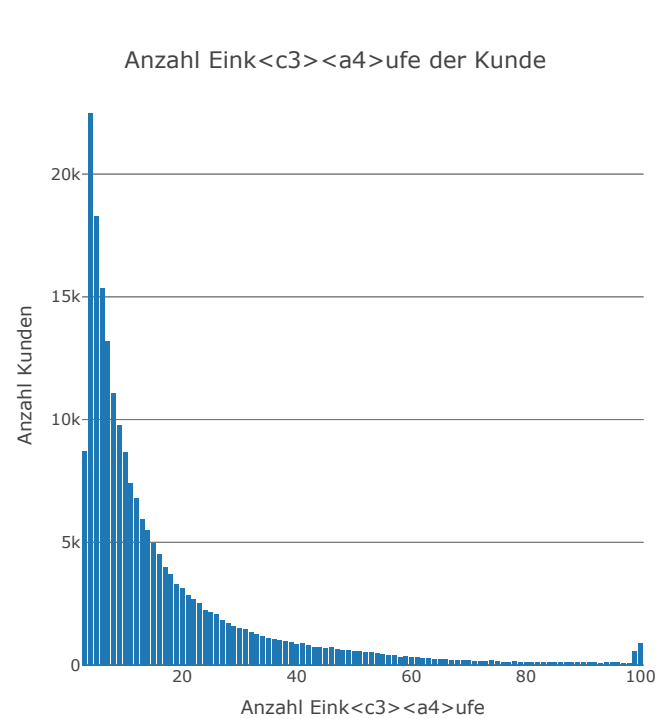
Produkte gekauft pro Bestellung pro Stunde pro Tag





Kundenzufriedenheit

Kommen die Kunden gerne zurück? In nachfolgenden Plot schauen wir uns an, ob es eine Kundenbindung gibt und somit ob die Kunden gerne diesen Service benutzen.



Die Kunden mögen Ihre Läden! Interessanterweise gibt es keinen einzigen Kunden, welcher nur einmal bei Ihnen einkaufen war. Dies liegt aber wohl eher daran, dass Warenkörbe mit nur einem oder zwei Produkten nicht in den Daten aufgenommen werden oder Instacart eine Mindestbestellanzahl hat. Ebenfalls auffallend ist der Cluster bei 99 und 100; Dies ist wohl ein Artefakt, welches sich zusammensetzt aus allen

Warenkörben mit mehr als 99, respektive 100 Produkten. Warum es zwei Cluster hat, ist uns nicht bekannt.

Wann kaufen Super-Stammkunden ein? Gibt es Unterschiede?

Momentan zählt jeder Kunde mit mehr als 50 Bestellungen als Super-Stammkunde.

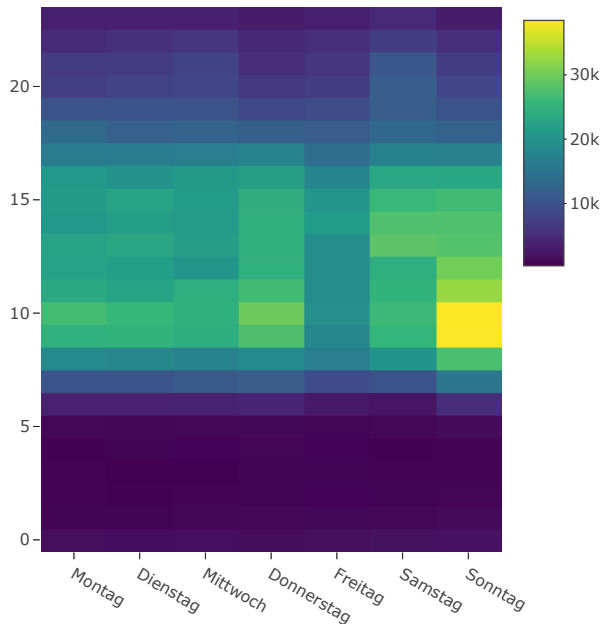
Code

Code

```
Joining, by = "order_dow"
```

Code

Bestellungen pro Stunde pro Tag von Super-Stammkunden



Interessant ist der Sonntag morgen, wo Stammkunden sehr häufig einkaufen gehen. ## Empfehlungen für Öffnungszeiten Wir geben Ihnen Anhand der vorliegenden Daten vier Empfehlungen:

1. Einfach und effizient

Von 8 Uhr Morgens bis 8 Uhr Abends. Dies deckt im Schnitt 88.3% der verkauften Produkte unter der Woche ab und zu 89.85% der verkauften Produkte am Wochenende. Diese Öffnungszeiten sind leicht zu merken und sozial. Trotzdem werden die Geschäfte geschlossen, sobald zuwenig eingekauft wird.

2. 22 Stunden pro Tag und 7 Tage offen für maximale Expansion

99.8% der verkauften Produkte abdecken? Vielleicht lässt sich so auch ein kleiner Tante-Emma Laden aus dem Markt drängen und noch mehr Profit herausholen.

3. Mehr Zeit zum Shoppen; Maximaler Verkauf

Öffnungszeiten von 7 bis 23 Uhr erlauben Ihnen unter der Woche und am Wochenende 97.4% der bestellten Produkte abzudecken.

4. Nur die wirklich wichtigen Stunden

Von 9 Uhr Morgens bis 15 Uhr Abends decken Sie immerhin noch 56.4% der Produkte ab unter der Woche und 58.2% am Wochenende.

Kurzes Schlusswort

Jede Empfehlung hat stärken und schwächen und in jeder Empfehlung kommen Super-Stammkunden zufrieden raus.

Weiteres Vorgehen

Zur Planung der Belastung könnte man auch nicht nur verkauften Produkte anschauen, sondern auch die Warenkorbgröße. Am Donnerstag zB. kommt ein Grosseinkauf, darauf könnte man sich speziell vorbereiten. Über Mittag sieht man auch, dass die Warenkörbe kleiner werden und die generierte Belastung von diesen könnten Self-Checkout Kassen übernehmen.

Code

```
Joining, by = "order_dow"
```

Code

Code

Produkte gekauft pro Bestellung



