

# Gruppe 4 - Final

## Daten einlesen

### Bereitgestellte Daten einlesen (im Hintergrund)

### Feiertagsdaten aus dem Netz holen (im Hintergrund)

## Daten zusammenfügen

```
## # A tibble: 10,899 x 11
##   Datum      Warengruppe Umsatz Wochentag   KielerWoche Bewoelkung Temperatur
##   <date>      <fct>      <dbl> <fct>      <dbl>      <dbl>      <dbl>
## 1 2013-07-01 Brot        149. Montag        NA          6        17.8
## 2 2013-07-02 Brot        160. Dienstag       NA          3        17.3
## 3 2013-07-03 Brot        112. Mittwoch        NA          7        21.1
## 4 2013-07-04 Brot        169. Donnerstag       NA          7        18.8
## 5 2013-07-05 Brot        171. Freitag         NA          5        20.0
## 6 2013-07-06 Brot        175. Samstag         NA          0        19.0
## 7 2013-07-07 Brot         92.6 Sonntag         NA          0        21.4
## 8 2013-07-08 Brot        136. Montag         NA          0        22.7
## 9 2013-07-09 Brot        136. Dienstag       NA          0        23.3
## 10 2013-07-10 Brot        135. Mittwoch        NA          2        19.7
## # ... with 10,889 more rows, and 4 more variables: Windgeschwindigkeit <dbl>,
## #   Wettercode <dbl>, Feiertage <chr>, istFeiertag <dbl>
```

## Datenaufbereitung

## Datenstruktur untersuchen

### Data summary

Name	dataset
Number of rows	10899
Number of columns	11
Column type frequency:	
character	1
Date	1
factor	2
numeric	7

Group variables

None

**Variable type: character**

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Feiertage	10784	0.01	11	25	0	5	0








**Variable type: Date**

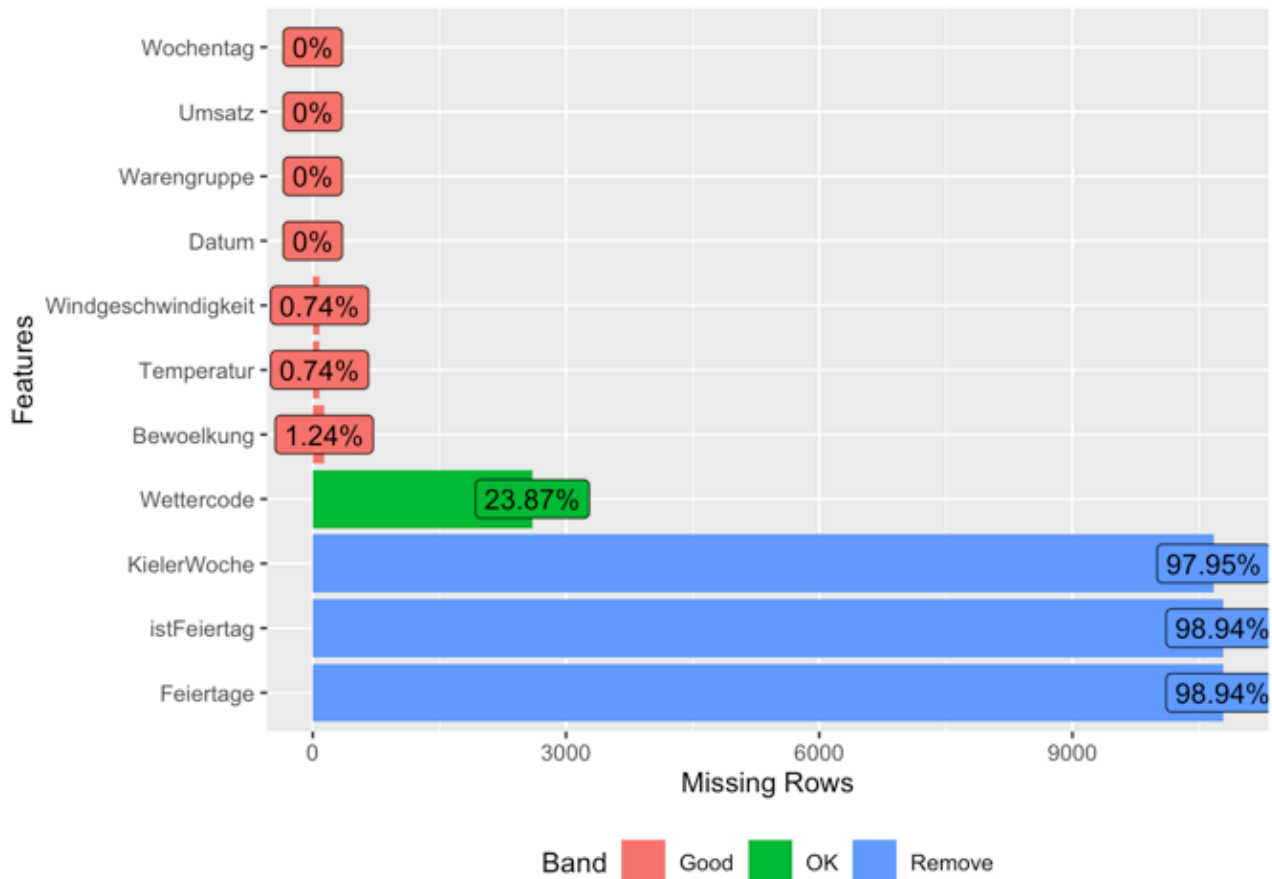
skim_variable	n_missing	complete_rate	min	max	median	n_unique
Datum	0	1	2013-07-01	2019-06-06	2016-06-17	2121

**Variable type: factor**

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Warengruppe	0	1	FALSE	6	Bro: 2121, Bro: 2121, Cro: 2121, Kuc: 2121
Wochentag	0	1	FALSE	7	Son: 1570, Don: 1568, Sam: 1564, Die: 1562

**Variable type: numeric**

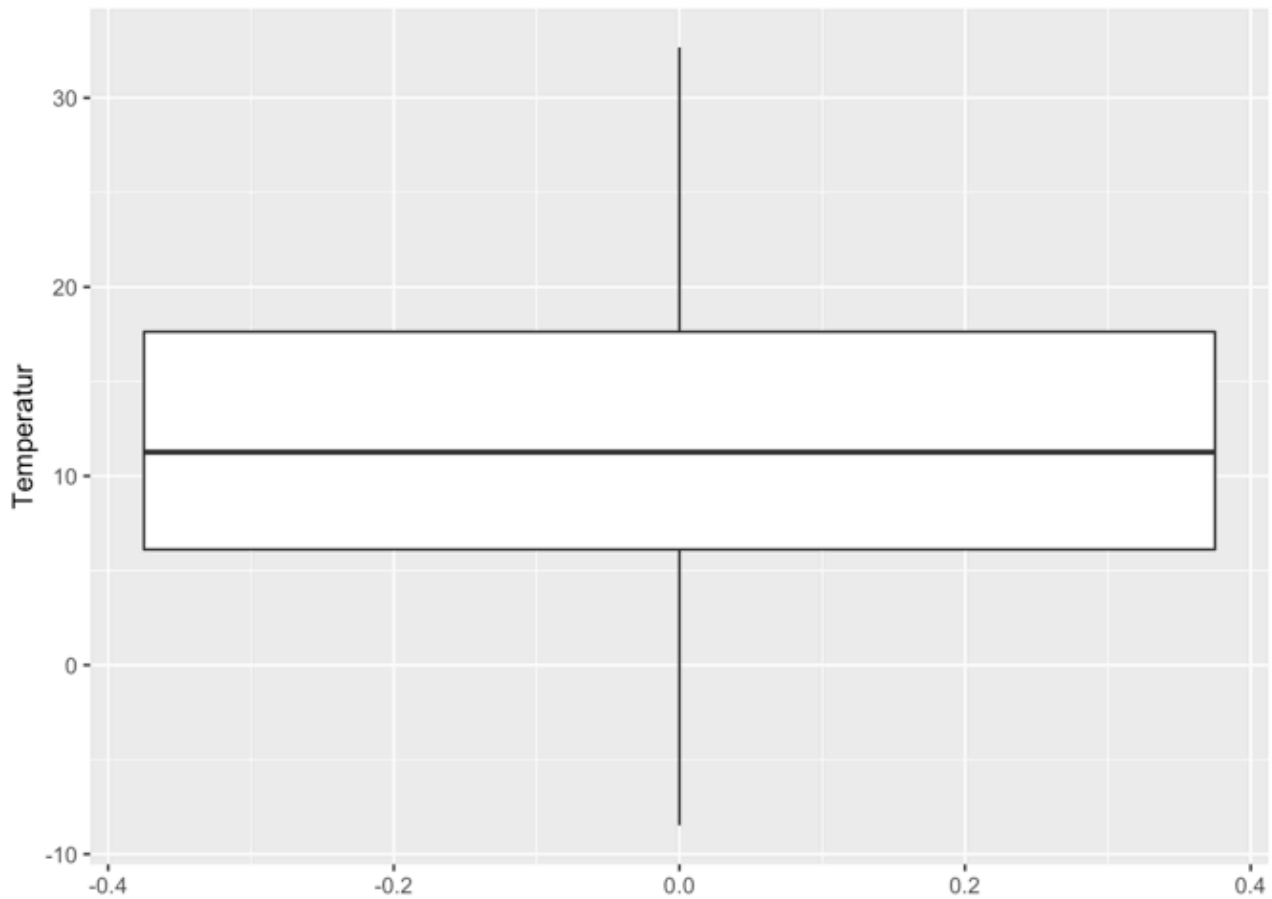
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Umsatz	0	1.00	206.66	142.81	7.05	97.53	163.30	280.81	1879.46	
KielerWoche	10676	0.02	1.00	0.00	1.00	1.00	1.00	1.00	1.00	
Bewoelkung	135	0.99	4.73	2.65	0.00	3.00	6.00	7.00	8.00	
Temperatur	81	0.99	11.82	7.15	-8.47	6.11	11.26	17.62	32.67	
Windgeschwindigkeit	81	0.99	11.00	4.14	3.00	8.00	10.00	13.00	35.00	
Wettercode	2602	0.76	36.16	27.04	0.00	10.00	22.00	61.00	95.00	
istFeiertag	10784	0.01	1.00	0.00	1.00	1.00	1.00	1.00	1.00	



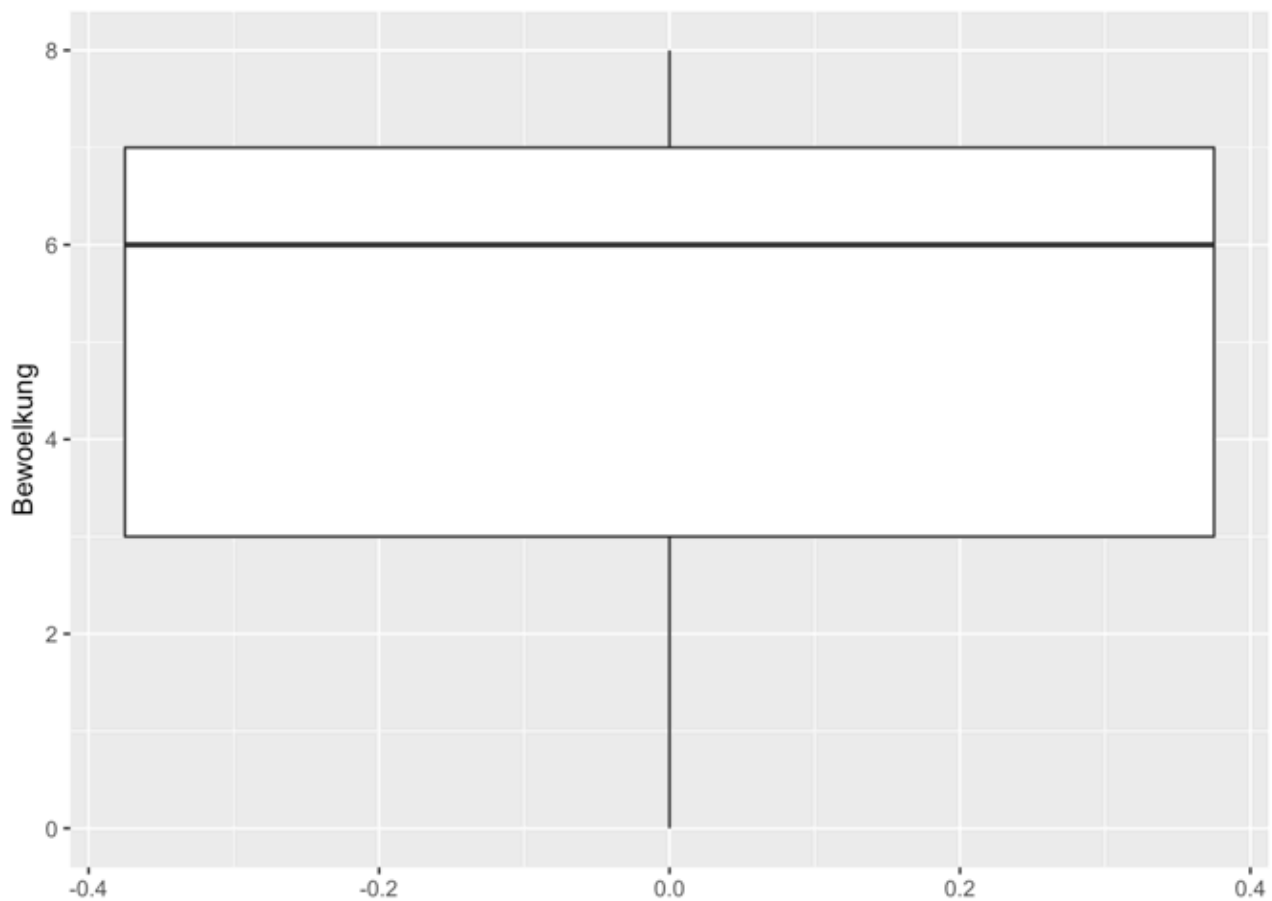
## Analyse der Ausreißer in den metrischen Variablen

- Temperatur: keine
- Bewoelkung: keine
- Windgeschwindigkeit: Ausreißer
- Umsatz: Ausreißer
- Umsatz nach Warengruppen: Ausreißer

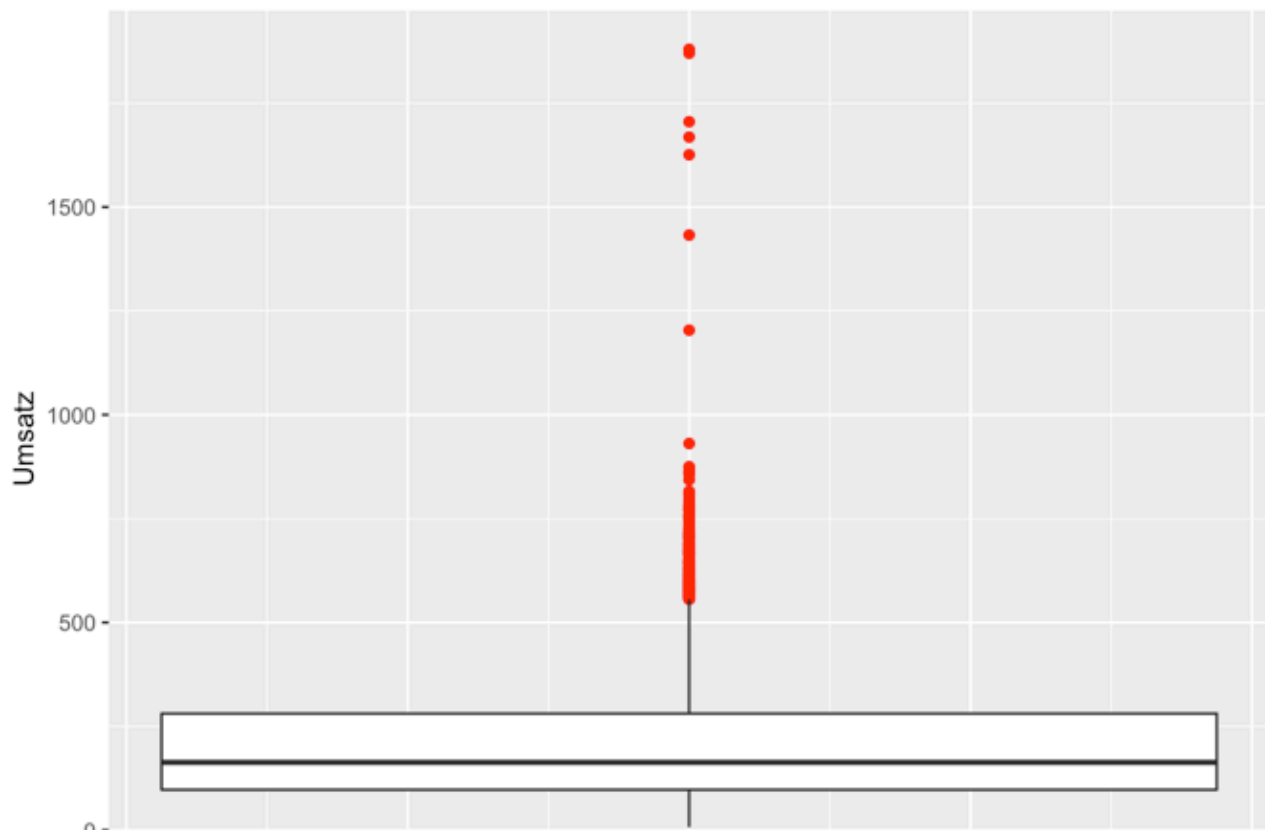
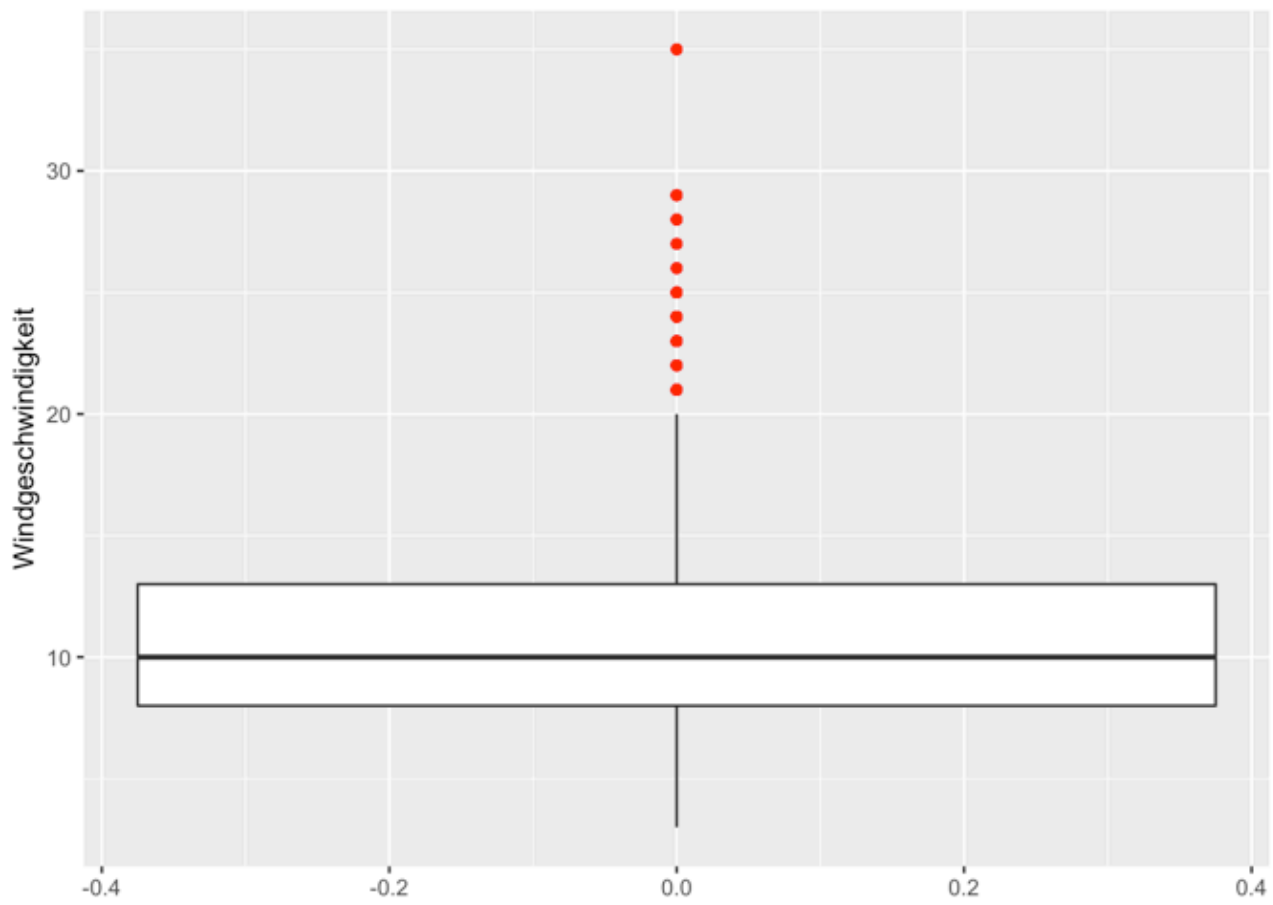
```
## Warning: Removed 81 rows containing non-finite values (stat_boxplot).
```

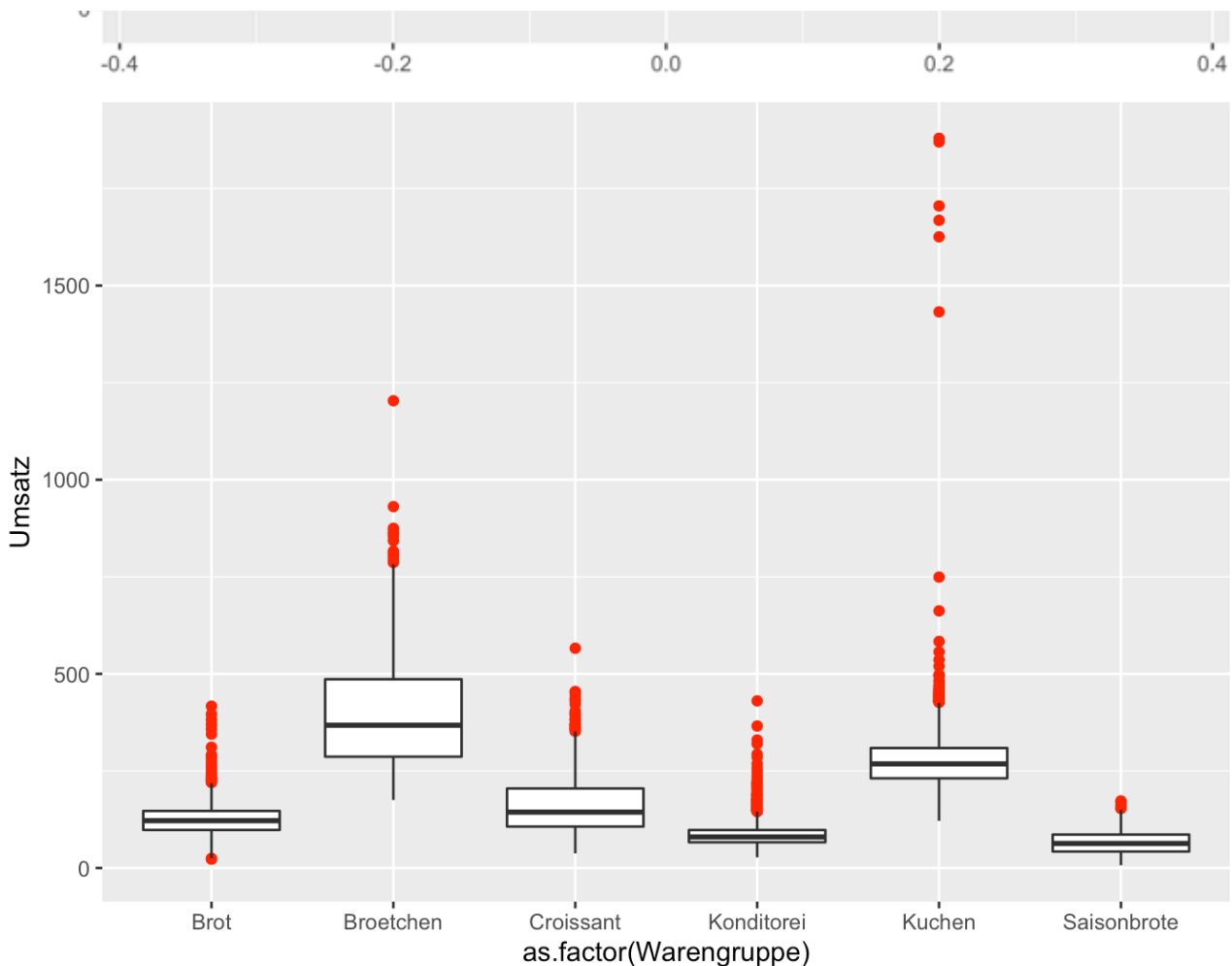


```
## Warning: Removed 135 rows containing non-finite values (stat_boxplot).
```



```
## Warning: Removed 81 rows containing non-finite values (stat_boxplot).
```





- Test auf Unterschiede der Mittelwerte im Umsatz per Warengruppen, um ggf. Warengruppen zusammenzufassen: nicht zu empfehlen, alle Unterschiede sig. von 0 verschieden

```
## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = Umsatz ~ Warengruppe, data = dataset)
##
## $Warengruppe
##
```

	diff	lwr	upr	p adj
Broetchen-Brot	274.47832	267.03128	281.92535	0.0000000
Croissant-Brot	40.11876	32.67173	47.56580	0.0000000
Konditorei-Brot	-36.80747	-44.30298	-29.31196	0.0000000
Kuchen-Brot	153.79543	146.34840	161.24246	0.0000000
Saisonbrote-Brot	-57.11407	-71.14025	-43.08790	0.0000000
Croissant-Broetchen	-234.35955	-241.80659	-226.91252	0.0000000
Konditorei-Broetchen	-311.28579	-318.78130	-303.79028	0.0000000
Kuchen-Broetchen	-120.68289	-128.12992	-113.23586	0.0000000
Saisonbrote-Broetchen	-331.59239	-345.61856	-317.56622	0.0000000
Konditorei-Croissant	-76.92624	-84.42175	-69.43072	0.0000000
Kuchen-Croissant	113.67666	106.22963	121.12370	0.0000000
Saisonbrote-Croissant	-97.23284	-111.25901	-83.20667	0.0000000
Kuchen-Konditorei	190.60290	183.10739	198.09841	0.0000000
Saisonbrote-Konditorei	-20.30660	-34.35858	-6.25463	0.0005478
Saisonbrote-Kuchen	-210.90950	-224.93568	-196.88333	0.0000000

# Datenaufbereitung: Fehlende Werte imputieren

- KielerWoche und istFeiertage dichotom (0 wenn nicht zutreffend)
- Bewoelkung, Temperatur und Windgeschwindigkeit mittlere Werte imputieren

## Data summary

Name	dataset
Number of rows	10899
Number of columns	11
Column type frequency:	
character	1
Date	1
factor	2
numeric	7
Group variables	
None	

## Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Feiertage	0	1	1	25	0	6	0




## Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
Datum	0	1	2013-07-01	2019-06-06	2016-06-17	2121

## Variable type: factor

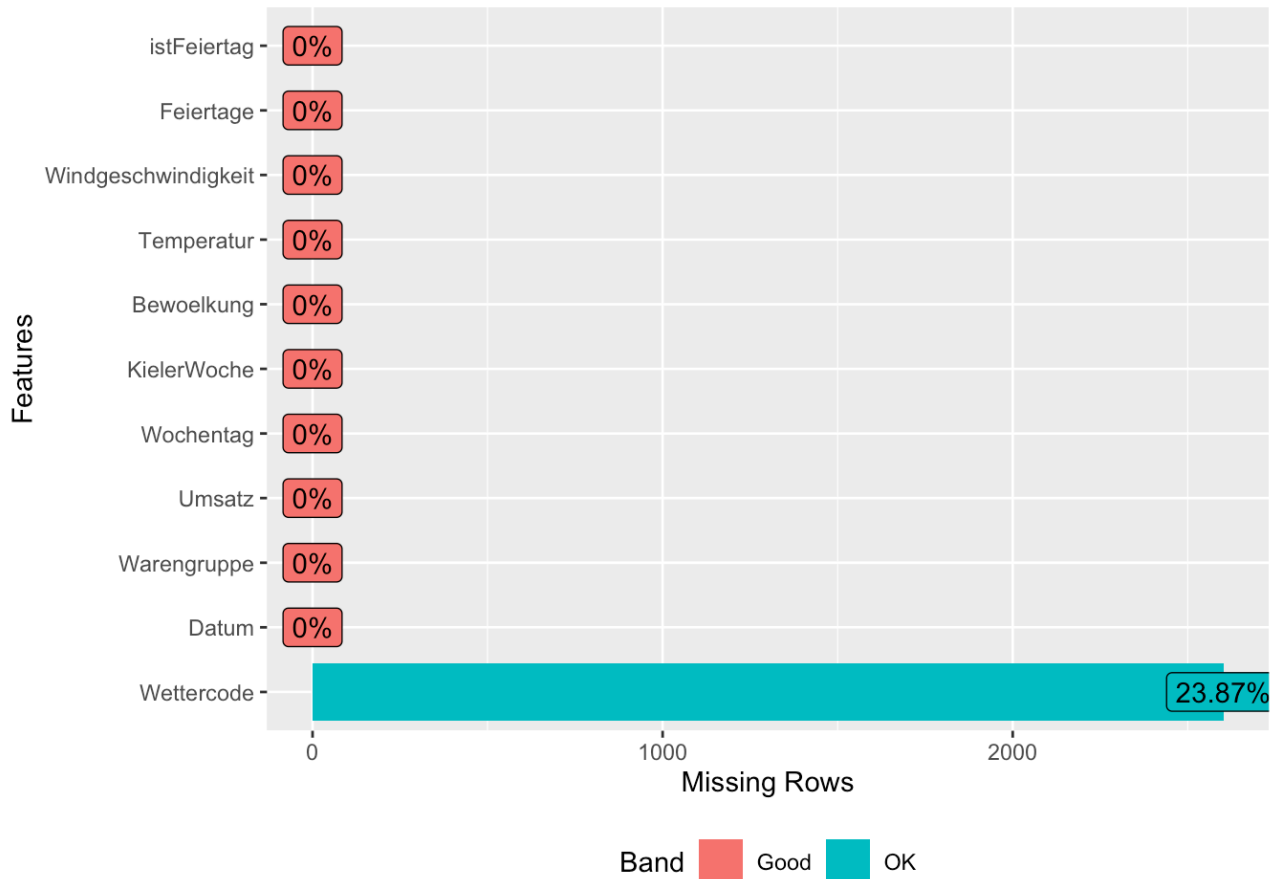
skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Warengruppe	0	1	FALSE	6	Bro: 2121, Bro: 2121, Cro: 2121, Kuc: 2121
Wochentag	0	1	FALSE	7	Son: 1570, Don: 1568, Sam: 1564, Die: 1562

## Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Umsatz	0	1.00	206.66	142.81	7.05	97.53	163.30	280.81	1879.46	
KielerWoche	0	1.00	0.02	0.14	0.00	0.00	0.00	0.00	1.00	
Bewoelkung	0	1.00	4.73	2.63	0.00	3.00	6.00	7.00	8.00	



Temperatur	0	1.00	11.82	7.13	-8.47	6.12	11.38	17.62	32.67	
Windgeschwindigkeit	0	1.00	11.00	4.12	3.00	8.00	10.00	13.00	35.00	
Wettercode	2602	0.76	36.16	27.04	0.00	10.00	22.00	61.00	95.00	
istFeiertag	0	1.00	0.01	0.10	0.00	0.00	0.00	0.00	1.00	



# Datensets für Modellierungen bereitstellen

## Trainingsdatensätze erstellen

- data1 : Faktoren bei Warengruppe und Wochentag bleiben erhalten, Variablen Wettercode und (Bezeichnung der) Feiertage raus
- data2 : Wie data1, aber alle Variablen numerisch

```
str(data1)
```

```
## tibble [10,899 × 9] (S3: tbl_df/tbl/data.frame)
## $ Datum          : Date[1:10899], format: "2013-07-01" "2013-07-02" ...
## $ Warengruppe    : Factor w/ 6 levels "Brot","Broetchen",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Umsatz         : num [1:10899] 149 160 112 169 171 ...
## $ Wochentag      : Factor w/ 7 levels "Montag","Dienstag",...: 1 2 3 4 5 6 7 1 2 3 ...
## $ KielerWoche    : num [1:10899] 0 0 0 0 0 0 0 0 0 0 ...
## $ Bewoelkung     : num [1:10899] 6 3 7 7 5 0 0 0 0 2 ...
## $ Temperatur     : num [1:10899] 17.8 17.3 21.1 18.9 20 ...
## $ Windgeschwindigkeit: num [1:10899] 15 10 6 7 12 8 9 10 8 13 ...
## $ istFeiertag    : num [1:10899] 0 0 0 0 0 0 0 0 0 0 ...
```

```
str(data2)
```

```
## tibble [10,899 × 9] (S3: tbl_df/tbl/data.frame)
## $ Datum          : Date[1:10899], format: "2013-07-01" "2013-07-02" ...
## $ Warengruppe    : num [1:10899] 1 1 1 1 1 1 1 1 1 1 ...
## $ Umsatz         : num [1:10899] 149 160 112 169 171 ...
## $ Wochentag      : num [1:10899] 1 2 3 4 5 6 7 1 2 3 ...
## $ KielerWoche    : num [1:10899] 0 0 0 0 0 0 0 0 0 0 ...
## $ Bewoelkung     : num [1:10899] 6 3 7 7 5 0 0 0 0 2 ...
## $ Temperatur     : num [1:10899] 17.8 17.3 21.1 18.9 20 ...
## $ Windgeschwindigkeit: num [1:10899] 15 10 6 7 12 8 9 10 8 13 ...
## $ istFeiertag    : num [1:10899] 0 0 0 0 0 0 0 0 0 0 ...
```

## Vorhersagedatensatz mit gleicher Struktur erzeugen

Prädiktion wird vorbereitet für Umsätze 7.6.2019 (Freitag, erster Tag nach Ende der Umsatzdaten) und 09.6.2019 (Sonntag darauf) und zum Kontrast ein stürmischer kalter Mittwoch im Januar 09.01.2019

- pred\_data1: Passend zu data1 mit Faktorenstrukturen
- pred\_data2: Passend zu data2 mit nur numerischen Variablen

```
str(pred_data1)
```

```
## 'data.frame':    18 obs. of  8 variables:
## $ Warengruppe    : Factor w/ 6 levels "Brot","Broetchen",...: 1 1 1 2 2 2 3 3 3 4 ...
## $ Datum          : Date, format: "2019-06-07" "2019-06-09" ...
## $ Wochentag      : Factor w/ 7 levels "Montag","Dienstag",...: 5 7 3 5 7 3 5 7 3 5 ...
## $ KielerWoche    : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Bewoelkung     : num  1 5 7 1 5 7 1 5 7 1 ...
## $ Temperatur     : num  17.45 18.65 3.14 17.45 18.65 ...
## $ Windgeschwindigkeit: num  10 10 22 10 10 22 10 10 22 10 ...
## $ istFeiertag    : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
str(pred_data2)
```

```
## 'data.frame':    18 obs. of  8 variables:
## $ Warengruppe      : num  1 1 1 2 2 2 3 3 3 4 ...
## $ Datum            : Date, format: "2019-06-07" "2019-06-09" ...
## $ Wochentag        : num  5 7 3 5 7 3 5 7 3 5 ...
## $ KielerWoche      : num  0 0 0 0 0 0 0 0 0 0 ...
## $ Bewoelkung       : num  1 5 7 1 5 7 1 5 7 1 ...
## $ Temperatur       : num  17.45 18.65 3.14 17.45 18.65 ...
## $ Windgeschwindigkeit: num  10 10 22 10 10 22 10 10 22 10 ...
## $ istFeiertag      : num  0 0 0 0 0 0 0 0 0 0 ...
```

# Lineare Modelle/Supervised Learning

## Optimierung des Modells mit schrittweiser Aufnahme von Variablen

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
(Intercept)	124.32 *** (1.85)	111.32 *** (2.61)	108.87 *** (2.62)	61.70 *** (2.74)	61.93 *** (2.74)	57.99 *** (3.38)	58.97 *** (3.84)
WarengruppeBroetchen	274.48 *** (2.61)	274.48 *** (2.51)	274.48 *** (2.50)	274.48 *** (2.34)	274.48 *** (2.34)	274.48 *** (2.34)	274.48 *** (2.34)
WarengruppeCroissant	40.12 *** (2.61)	40.12 *** (2.51)	40.12 *** (2.50)	40.12 *** (2.34)	40.12 *** (2.34)	40.12 *** (2.34)	40.12 *** (2.34)
WarengruppeKonditorei	-36.81 *** (2.63)	-36.97 *** (2.52)	-36.99 *** (2.51)	-36.84 *** (2.36)	-36.83 *** (2.36)	-36.82 *** (2.36)	-36.82 *** (2.36)
WarengruppeKuchen	153.80 *** (2.61)	153.80 *** (2.51)	153.80 *** (2.50)	153.80 *** (2.34)	153.80 *** (2.34)	153.80 *** (2.34)	153.80 *** (2.34)
WarengruppeSaisonbrote	-57.11 *** (4.92)	-57.05 *** (4.72)	-56.30 *** (4.71)	-35.08 *** (4.45)	-34.96 *** (4.45)	-34.68 *** (4.45)	-34.62 *** (4.45)
WochentagDienstag		-6.23 * (2.93)	-4.02 (2.93)	-4.76 (2.75)	-4.74 (2.74)	-4.75 (2.74)	-4.72 (2.74)
WochentagMittwoch		-7.31 * (2.93)	-5.09 (2.93)	-6.29 * (2.75)	-6.26 * (2.75)	-6.28 * (2.75)	-6.28 * (2.75)
WochentagDonnerstag		0.70	1.59	0.57	0.59	0.67	0.71

	(2.92)	(2.92)	(2.73)	(2.73)	(2.73)	(2.73)
WochentagFreitag	2.78	4.99	3.65	3.68	3.75	3.77
	(2.94)	(2.94)	(2.76)	(2.76)	(2.76)	(2.76)
WochentagSamstag	47.36 ***	49.57 ***	49.64 ***	49.41 ***	49.50 ***	49.53 ***
	(2.93)	(2.93)	(2.74)	(2.74)	(2.74)	(2.74)
WochentagSonntag	53.24 ***	55.45 ***	54.69 ***	54.46 ***	54.47 ***	54.46 ***
	(2.92)	(2.92)	(2.74)	(2.74)	(2.74)	(2.74)
istFeiertag		68.58 ***	61.46 ***	61.83 ***	61.49 ***	61.33 ***
		(7.68)	(7.21)	(7.21)	(7.21)	(7.21)
Temperatur			4.00 ***	3.95 ***	3.96 ***	3.93 ***
			(0.10)	(0.10)	(0.10)	(0.11)
KielerWoche				16.09 **	15.90 **	16.09 **
				(5.22)	(5.22)	(5.23)
Windgeschwindigkeit					0.35 *	0.36 *
					(0.18)	(0.18)
Bewoelkung						-0.16
						(0.30)
N	10899	10899	10899	10899	10899	10899
R2	0.65	0.67	0.68	0.71	0.72	0.72

\*\*\* p < 0.001; \*\* p < 0.01; \* p < 0.05.

```
## Analysis of Variance Table
##
## Model 1: Umsatz ~ Warengruppe
## Model 2: Umsatz ~ Warengruppe + Wochentag
## Model 3: Umsatz ~ Warengruppe + Wochentag + istFeiertag
## Model 4: Umsatz ~ Warengruppe + Wochentag + istFeiertag + Temperatur
## Model 5: Umsatz ~ Warengruppe + Wochentag + istFeiertag + Temperatur +
##           KielerWoche
## Model 6: Umsatz ~ Warengruppe + Wochentag + istFeiertag + Temperatur +
##           KielerWoche + Windgeschwindigkeit
## Model 7: Umsatz ~ Warengruppe + Wochentag + istFeiertag + Temperatur +
##           KielerWoche + Windgeschwindigkeit + Bewoelkung
##   Res.Df      RSS Df Sum of Sq      F      Pr(>F)
## 1  10893 78862365
## 2  10887 72598376  6   6263989  179.4584 < 2.2e-16 ***
## 3  10886 72070850  1    527526   90.6793 < 2.2e-16 ***
## 4  10885 63386080  1   8684769 1492.8712 < 2.2e-16 ***
## 5  10884 63330707  1    55373    9.5184  0.002039 **
## 6  10883 63307669  1    23038    3.9601  0.046615 *
## 7  10882 63305971  1     1698    0.2919  0.589009
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Entscheidung für das Modell 6 **Umsatz ~ Warengruppe + Wochentag + istFeiertag + Temperatur + KielerWoche + Windgeschwindigkeit** mit 0,715 Varianzaufklärung

```
## MODEL INFO:
## Observations: 10899
## Dependent Variable: Umsatz
## Type: OLS linear regression
##
## MODEL FIT:
## F(15,10883) = 1821.72, p = 0.00
## R2 = 0.72
## Adj. R2 = 0.71
##
## Standard errors: OLS
## -----
##                               Est.    S.E.    t val.    p
## -----
## (Intercept)                57.99    3.38     17.15    0.00
## WarengruppeBroetchen       274.48    2.34    117.20    0.00
## WarengruppeCroissant        40.12    2.34     17.13    0.00
## WarengruppeKonditorei      -36.82    2.36    -15.62    0.00
## WarengruppeKuchen          153.80    2.34     65.67    0.00
## WarengruppeSaisonbrote     -34.68    4.45     -7.79    0.00
## WochentagDienstag           -4.75    2.74     -1.73    0.08
## WochentagMittwoch           -6.28    2.75     -2.28    0.02
## WochentagDonnerstag         0.67    2.73      0.25    0.81
## WochentagFreitag            3.75    2.76      1.36    0.17
## WochentagSamstag           49.50    2.74     18.04    0.00
## WochentagSonntag           54.47    2.74     19.87    0.00
## istFeiertag                61.49    7.21      8.53    0.00
## Temperatur                  3.96    0.10     37.90    0.00
## KielerWoche                15.90    5.22      3.05    0.00
## Windgeschwindigkeit         0.35    0.18      1.99    0.05
## -----
```

## Prädiktion 1 (Lineare Modelle/Supervised Learning)

aka: Ich nehme ein lineares Modell, das auf einen Datensatz gefittet wurde und sage mit den dort geschätzten Koeffizienten die Zielvariable auf Basis von einem neuen Datensatz vorher

```

#Vorhersage berechnen
pred_lm <- predict(object=LMfinal, newdata = pred_data1, type = "response", interval = "confidence", na.action = na.exclude)

#print(pred_lm)
#class(pred_lm)#is ne doofe Matrix--> für Nutzung von dplyr-Routinen as.data.frame()

pred_lm <- dplyr::bind_cols(pred_data1, as.data.frame(pred_lm)) #in dieser Datei sind "fit" die Umsatz-Prädiktionen für die jeweiligen Datenpunkte, lwr und upr sind die Grenzen der Konfidenzintervalle

pred_lm<- rename(pred_lm,Prediction_Umsatz_LM = fit,UntereGrenzeKonfInt=lwr,ObereGrenzeKonfInt=upr)

anzeige <-dplyr::filter(pred_lm,Datum==ymd(20190607))
anzeige <-dplyr::select(anzeige, c("Warengruppe","Prediction_Umsatz_LM","UntereGrenzeKonfInt","ObereGrenzeKonfInt"))
pred_lm

```

##	Warengruppe	Datum	Wochentag	KielerWoche	Bewoelkung	Temperatur
## 1	Brot	2019-06-07	Freitag	0	1	17.4500
## 2	Brot	2019-06-09	Sonntag	0	5	18.6500
## 3	Brot	2019-01-09	Mittwoch	0	7	3.1375
## 4	Broetchen	2019-06-07	Freitag	0	1	17.4500
## 5	Broetchen	2019-06-09	Sonntag	0	5	18.6500
## 6	Broetchen	2019-01-09	Mittwoch	0	7	3.1375
## 7	Croissant	2019-06-07	Freitag	0	1	17.4500
## 8	Croissant	2019-06-09	Sonntag	0	5	18.6500
## 9	Croissant	2019-01-09	Mittwoch	0	7	3.1375
## 10	Konditorei	2019-06-07	Freitag	0	1	17.4500
## 11	Konditorei	2019-06-09	Sonntag	0	5	18.6500
## 12	Konditorei	2019-01-09	Mittwoch	0	7	3.1375
## 13	Kuchen	2019-06-07	Freitag	0	1	17.4500
## 14	Kuchen	2019-06-09	Sonntag	0	5	18.6500
## 15	Kuchen	2019-01-09	Mittwoch	0	7	3.1375
## 16	Saisonbrote	2019-06-07	Freitag	0	1	17.4500
## 17	Saisonbrote	2019-06-09	Sonntag	0	5	18.6500
## 18	Saisonbrote	2019-01-09	Mittwoch	0	7	3.1375
##	Windgeschwindigkeit	istFeiertag	Prediction_Umsatz_LM	UntereGrenzeKonfInt		
## 1	10	0	134.29212	129.35192		
## 2	10	0	189.76167	184.76491		
## 3	22	0	71.89466	65.53845		
## 4	10	0	408.77044	403.83024		
## 5	10	0	464.23999	459.24322		
## 6	22	0	346.37298	340.01676		
## 7	10	0	174.41089	169.47068		
## 8	10	0	229.88044	224.88367		
## 9	22	0	112.01343	105.65721		
## 10	10	0	97.46888	92.49775		
## 11	10	0	152.93843	147.92204		
## 12	22	0	35.07142	28.68628		
## 13	10	0	288.08755	283.14735		
## 14	10	0	343.55710	338.56034		
## 15	22	0	225.69009	219.33387		

## 16	10	0	99.61699	90.59195
## 17	10	0	155.08654	146.00653
## 18	22	0	37.21953	27.53456
##	ObereGrenzeKonfInt			
## 1	139.23233			
## 2	194.75844			
## 3	78.25088			
## 4	413.71065			
## 5	469.23675			
## 6	352.72920			
## 7	179.35109			
## 8	234.87720			
## 9	118.36964			
## 10	102.44001			
## 11	157.95481			
## 12	41.45656			
## 13	293.02776			
## 14	348.55387			
## 15	232.04631			
## 16	108.64204			
## 17	164.16655			
## 18	46.90450			

**Die Vorhersage für den ersten Tag (2019-06-07) nach Ende der Datenreihe ist:**

##	Warengruppe	Prediction_Umsatz_LM	UntereGrenzeKonfInt	ObereGrenzeKonfInt
## 1	Brot	134.29212	129.35192	139.2323
## 4	Broetchen	408.77044	403.83024	413.7106
## 7	Croissant	174.41089	169.47068	179.3511
## 10	Konditorei	97.46888	92.49775	102.4400
## 13	Kuchen	288.08755	283.14735	293.0278
## 16	Saisonbrote	99.61699	90.59195	108.6420

# Support Vector Maschinen

## Trainings- und Testdatensätze erstellen

**Achtung: Im Moment auf 10% Trainingsdaten, wegen Performanz! Am Ende wieder anpassen**



```
data<-data2

# Zufallszähler setzen (um die zufällige Partitionierung bei jedem Durchlauf gleich zu halten
)
set.seed(1)

#Datensatzzeilen shuffeln
new_row_order<-sample(nrow(data))
data<-data[new_row_order,]

# Zufällige Ziehung von Indizes für die Zeilen des Datensatzes, die dem Trainingsdatensatz zug
eordnet werden, Umfang: 80% (im Moment 10%)
indices_train <- sample(seq_len(nrow(data)), size = floor(0.10 * nrow(data)))

# Definition des Trainings- und Testdatensatz durch Selektion bzw. Deselektion der entspreche
nden Datenzeilen
train_data <- data[indices_train, ]
test_data <- data[-indices_train, ]
```

## SVM Trainieren (ohne Tuning)

```
model_svm <- svm(Umsatz ~ Warengruppe+Wochentag, train_data)
summary(model_svm)
```

```
##
## Call:
## svm(formula = Umsatz ~ Warengruppe + Wochentag, data = train_data)
##
##
## Parameters:
##   SVM-Type:  eps-regression
##   SVM-Kernel: radial
##         cost:  1
##        gamma: 0.5
##      epsilon: 0.1
##
##
## Number of Support Vectors:  801
```

## Kreuzvalidierung

```
test_data_svm <-predict(model_svm,test_data,na.action = na.pass)
mape(test_data$Umsatz,test_data_svm) #Mean Absolute Percent Error,
```

```
## [1] 0.2777463
```

```
#rse: sum(error^2)/sum(actual-mean(actual))  
#r^2=1-rse #laut Steffen Brandt  
  
r_squared_svm<-1-rse(test_data$Umsatz,test_data_svm)  
r_squared_svm
```

```
## [1] 0.6697342
```

### ##Prädiktion 2 (SVM ohne Tuning)

```
pred_svm <- predict(model_svm, pred_data2, na.action = na.pass)# Daten Vorhersagen  
  
pred_svm <- dplyr::bind_cols(pred_data2, as.data.frame(pred_svm )) #zusammenfügen  
  
pred_svm<- rename(pred_svm,Prediction_Umsatz_SVM = pred_svm)#Variablenbenennung anpassen  
  
#Faktoren wieder schick machen  
pred_svm$Wochentag <- factor(weekdays(pred_svm$Datum),  
levels=ordentlicheWoche)  
pred_svm$Warengruppe <- factor(pred_svm$Warengruppe,  
levels = c(1,2,3,4,5,6),  
labels = c("Brot", "Broetchen", "Croissant", "Konditorei", "Kuchen", "Saisonbrote"))  
  
anzeige <-dplyr::filter(pred_svm,Datum==ymd(20190607))  
anzeige <-dplyr::select(anzeige, c("Warengruppe","Prediction_Umsatz_SVM"))  
  
pred_svm
```

##	Warengruppe	Datum	Wochentag	KielerWoche	Bewoelkung	Temperatur
## 1	Brot	2019-06-07	Freitag	0	1	17.4500
## 2	Brot	2019-06-09	Sonntag	0	5	18.6500
## 3	Brot	2019-01-09	Mittwoch	0	7	3.1375
## 4	Broetchen	2019-06-07	Freitag	0	1	17.4500
## 5	Broetchen	2019-06-09	Sonntag	0	5	18.6500
## 6	Broetchen	2019-01-09	Mittwoch	0	7	3.1375
## 7	Croissant	2019-06-07	Freitag	0	1	17.4500
## 8	Croissant	2019-06-09	Sonntag	0	5	18.6500
## 9	Croissant	2019-01-09	Mittwoch	0	7	3.1375
## 10	Konditorei	2019-06-07	Freitag	0	1	17.4500
## 11	Konditorei	2019-06-09	Sonntag	0	5	18.6500
## 12	Konditorei	2019-01-09	Mittwoch	0	7	3.1375
## 13	Kuchen	2019-06-07	Freitag	0	1	17.4500
## 14	Kuchen	2019-06-09	Sonntag	0	5	18.6500
## 15	Kuchen	2019-01-09	Mittwoch	0	7	3.1375
## 16	Saisonbrote	2019-06-07	Freitag	0	1	17.4500
## 17	Saisonbrote	2019-06-09	Sonntag	0	5	18.6500
## 18	Saisonbrote	2019-01-09	Mittwoch	0	7	3.1375

##	Windgeschwindigkeit	istFeiertag	Prediction_Umsatz_SVM
## 1	10	0	139.46187
## 2	10	0	102.32115
## 3	22	0	120.07940
## 4	10	0	373.01217
## 5	10	0	444.25262
## 6	22	0	324.04645
## 7	10	0	146.34016
## 8	10	0	245.29885
## 9	22	0	125.26464
## 10	10	0	76.18348
## 11	10	0	124.71059
## 12	22	0	75.99692
## 13	10	0	261.05519
## 14	10	0	276.48536
## 15	22	0	253.58632
## 16	10	0	92.94658
## 17	10	0	154.84027
## 18	22	0	76.55080

Die Vorhersage für den ersten Tag (2019-06-07) nach Ende der Datenreihe ist:

##	Warengruppe	Prediction_Umsatz_SVM
## 1	Brot	139.46187
## 4	Broetchen	373.01217
## 7	Croissant	146.34016
## 10	Konditorei	76.18348
## 13	Kuchen	261.05519
## 16	Saisonbrote	92.94658

## SVM Trainieren (mit Tuning)

```
svm_tune <- tune(svm, Umsatz ~ Warengruppe + Wochentag, data=train_data, ranges = list(epsilon
= seq(0.2,1,0.1), cost = 2^(2:3)))
summary(svm_tune)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##   epsilon cost
##     0.3      8
##
## - best performance: 4952.79
##
## - Detailed performance results:
##   epsilon cost   error dispersion
## 1      0.2      4 5026.049    1820.516
## 2      0.3      4 5006.029    1788.866
## 3      0.4      4 5120.673    1825.155
## 4      0.5      4 5283.028    1781.354
## 5      0.6      4 5504.911    1672.792
## 6      0.7      4 5875.760    1578.470
## 7      0.8      4 6387.392    1495.979
## 8      0.9      4 7226.653    1495.778
## 9      1.0      4 8180.528    1397.576
## 10     0.2      8 4957.881    1797.609
## 11     0.3      8 4952.790    1777.738
## 12     0.4      8 5037.437    1782.091
## 13     0.5      8 5177.292    1756.901
## 14     0.6      8 5391.655    1633.959
## 15     0.7      8 5794.482    1528.910
## 16     0.8      8 6252.852    1456.429
## 17     0.9      8 6889.821    1383.743
## 18     1.0      8 7711.824    1317.151
```

## Kreuzvalidierung

```
test_data_svm<-predict(svm_tune$best.model,test_data,na.action = na.pass)
mape(test_data$Umsatz,test_data_svm) #Mean Absolute Percent Error
```

```
## [1] 0.2787312
```

```
r_squared_svmtune<-1-rse(test_data$Umsatz,test_data_svm)
r_squared_svmtune
```

```
## [1] 0.689356
```

## Prädiktion 3 (SVM mit Tuning)

```
pred_svm_tune <- predict(svm_tune$best.model, pred_data2, na.action = na.pass) # Daten Vorhersagen  
  
pred_svm_tune <- dplyr::bind_cols(pred_data2, as.data.frame(pred_svm_tune)) # zusammenfügen  
  
pred_svm_tune <- rename(pred_svm_tune, Prediction_Umsatz_SVM_tune = pred_svm_tune) # Variablenbenennung anpassen  
  
# Faktoren wieder schick machen  
pred_svm_tune$Wochentag <- factor(weekdays(pred_svm_tune$Datum),  
  levels = ordentlicheWoche)  
pred_svm_tune$Warengruppe <- factor(pred_svm_tune$Warengruppe,  
  levels = c(1, 2, 3, 4, 5, 6),  
  labels = c("Brot", "Broetchen", "Croissant", "Konditorei", "Kuchen", "Saisonbrote"))  
  
anzeige <- dplyr::filter(pred_svm_tune, Datum == ymd(20190607))  
anzeige <- dplyr::select(anzeige, c("Warengruppe", "Prediction_Umsatz_SVM_tune"))  
  
pred_svm_tune
```

```
##      Warengruppe      Datum Wochentag KielerWoche Bewoelkung Temperatur
## 1      Brot 2019-06-07   Freitag          0          1      17.4500
## 2      Brot 2019-06-09   Sonntag          0          5      18.6500
## 3      Brot 2019-01-09  Mittwoch          0          7       3.1375
## 4    Broetchen 2019-06-07   Freitag          0          1      17.4500
## 5    Broetchen 2019-06-09   Sonntag          0          5      18.6500
## 6    Broetchen 2019-01-09  Mittwoch          0          7       3.1375
## 7    Croissant 2019-06-07   Freitag          0          1      17.4500
## 8    Croissant 2019-06-09   Sonntag          0          5      18.6500
## 9    Croissant 2019-01-09  Mittwoch          0          7       3.1375
## 10 Konditorei 2019-06-07   Freitag          0          1      17.4500
## 11 Konditorei 2019-06-09   Sonntag          0          5      18.6500
## 12 Konditorei 2019-01-09  Mittwoch          0          7       3.1375
## 13      Kuchen 2019-06-07   Freitag          0          1      17.4500
## 14      Kuchen 2019-06-09   Sonntag          0          5      18.6500
## 15      Kuchen 2019-01-09  Mittwoch          0          7       3.1375
## 16 Saisonbrote 2019-06-07   Freitag          0          1      17.4500
## 17 Saisonbrote 2019-06-09   Sonntag          0          5      18.6500
## 18 Saisonbrote 2019-01-09  Mittwoch          0          7       3.1375
##      Windgeschwindigkeit istFeiertag Prediction_Umsatz_SVMtune
## 1              10          0          143.95391
## 2              10          0          100.17056
## 3              22          0          121.56011
## 4              10          0          377.52470
## 5              10          0          488.12536
## 6              22          0          346.08386
## 7              10          0          144.56291
## 8              10          0          217.06357
## 9              22          0          139.47797
## 10             10          0           84.78224
## 11             10          0          120.85141
## 12             22          0           82.56540
## 13             10          0          270.96007
## 14             10          0          304.77103
## 15             22          0          261.54012
## 16             10          0           89.00212
## 17             10          0          108.28671
## 18             22          0           71.69819
```

**Die Vorhersage für den ersten Tag (2019-06-07) nach Ende der Datenreihe ist:**

```
##      Warengruppe Prediction_Umsatz_SVMtune
## 1      Brot          143.95391
## 4    Broetchen          377.52470
## 7    Croissant          144.56291
## 10 Konditorei           84.78224
## 13      Kuchen          270.96007
## 16 Saisonbrote           89.00212
```

```
## Warning in rm(anzeige, indices_train, new_row_order, test_data_svm): Objekt
## 'indices_train' nicht gefunden
```

```
## Warning in rm(anzeige, indices_train, new_row_order, test_data_svm): Objekt
## 'new_row_order' nicht gefunden
```

# Hier dann noch die Neuronalen Netze etc. einfügen

## Grafischer Vergleich der Vorhersagen

##	Warengruppe	Datum	Wochentag	KielerWoche	Bewoelkung	Temperatur
## 1	Brot	2019-06-07	Freitag	0	1	17.4500
## 2	Brot	2019-06-09	Sonntag	0	5	18.6500
## 3	Brot	2019-01-09	Mittwoch	0	7	3.1375
## 4	Broetchen	2019-06-07	Freitag	0	1	17.4500
## 5	Broetchen	2019-06-09	Sonntag	0	5	18.6500
## 6	Broetchen	2019-01-09	Mittwoch	0	7	3.1375
## 7	Croissant	2019-06-07	Freitag	0	1	17.4500
## 8	Croissant	2019-06-09	Sonntag	0	5	18.6500
## 9	Croissant	2019-01-09	Mittwoch	0	7	3.1375
## 10	Konditorei	2019-06-07	Freitag	0	1	17.4500
## 11	Konditorei	2019-06-09	Sonntag	0	5	18.6500
## 12	Konditorei	2019-01-09	Mittwoch	0	7	3.1375
## 13	Kuchen	2019-06-07	Freitag	0	1	17.4500
## 14	Kuchen	2019-06-09	Sonntag	0	5	18.6500
## 15	Kuchen	2019-01-09	Mittwoch	0	7	3.1375
## 16	Saisonbrote	2019-06-07	Freitag	0	1	17.4500
## 17	Saisonbrote	2019-06-09	Sonntag	0	5	18.6500
## 18	Saisonbrote	2019-01-09	Mittwoch	0	7	3.1375
##	Windgeschwindigkeit	istFeiertag	Prediction_Umsatz_LM	Prediction_Umsatz_SVM		
## 1	10	0	134.29212	139.46187		
## 2	10	0	189.76167	102.32115		
## 3	22	0	71.89466	120.07940		
## 4	10	0	408.77044	373.01217		
## 5	10	0	464.23999	444.25262		
## 6	22	0	346.37298	324.04645		
## 7	10	0	174.41089	146.34016		
## 8	10	0	229.88044	245.29885		
## 9	22	0	112.01343	125.26464		
## 10	10	0	97.46888	76.18348		
## 11	10	0	152.93843	124.71059		
## 12	22	0	35.07142	75.99692		
## 13	10	0	288.08755	261.05519		
## 14	10	0	343.55710	276.48536		
## 15	22	0	225.69009	253.58632		
## 16	10	0	99.61699	92.94658		
## 17	10	0	155.08654	154.84027		
## 18	22	0	37.21953	76.55080		
##	Prediction_Umsatz_SVMtune					
## 1	143.95391					
## 2	100.17056					
## 3	121.56011					
## 4	377.52470					
## 5	488.12536					
## 6	346.08386					
## 7	144.56291					
## 8	217.06357					

## 9	139.47797
## 10	84.78224
## 11	120.85141
## 12	82.56540
## 13	270.96007
## 14	304.77103
## 15	261.54012
## 16	89.00212
## 17	108.28671
## 18	71.69819



