

Fachhochschule Südwestfalen  
Fachbereich Ingenieur- und Wirtschaftswissenschaften

Forecasting (Prof. Dr. Buchwitz)

Gutachter: Prof. Dr. Buchwitz

# Fallstudie über Verkehrsdaten von PKW, LKW und Bussen mit Naiven, ETS und ARIMA Modellen

E B

---

## Zusammenfassung

In dieser Fallstudie werden die Zeitreihenstrukturen der stündlich erfassten Verkehrsdaten aus dem Jahr 2022 aus Meschede von der Bundesanstalt für Verkehrswesen analysiert. Dafür werden die drei Fahrzeugklassen PKW, LKW und Busse genauer betrachtet. Es werden Naive Methoden, ETS Modelle und ARIMA Modelle verglichen. ARIMA Modelle erfassen die Strukturen der Verkehrsdaten von LKW und Bussen gut, erfassen jedoch die Strukturen der PKW nicht ausreichend, ebenso wie die ETS Modelle

**Keywords:** Statistik, Regression, Forecasting, Verkehrsdaten, ARIMA, ETS

---

Meschede  
3. September 2025

# Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbständig angefertigt habe. Die aus fremden Quellen direkt und indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde weder einer anderen Prüfungsbehörde vorgelegt noch veröffentlicht.

Ich weiß, dass die Arbeit in digitalisierter Form daraufhin überprüft werden kann, ob unerlaubte Hilfsmittel verwendet wurden und ob es sich – insgesamt oder in Teilen – um ein Plagiat handelt. Zum Vergleich meiner Arbeit mit existierenden Quellen darf sie in eine Datenbank eingestellt werden und nach der Überprüfung zum Vergleich mit künftig eingehenden Arbeiten dort verbleiben.

Meschede, 3. September 2025.

**E B**

MatNr: 30360314

Email: {}

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>4</b>
1.1	Wofür? . . . . .	4
1.2	Was? . . . . .	4
<b>2</b>	<b>Daten</b>	<b>4</b>
2.1	Laden und Aufbereitung der Daten . . . . .	4
2.2	Datenaufbereitung für Klassen . . . . .	10
2.3	Beobachtungen aus 5+1 auf Tagesebene: . . . . .	14
2.4	Überlegungen für weiteres Vorgehen . . . . .	14
<b>3</b>	<b>Benchmarks</b>	<b>17</b>
3.1	Trainings-/Testsplit . . . . .	17
3.2	Naive Methoden . . . . .	17
3.3	Auswertung . . . . .	24
3.4	Anmerkung . . . . .	25
<b>4</b>	<b>ETS</b>	<b>25</b>
4.1	Theretische Überlegungen . . . . .	25
4.2	Praxis . . . . .	25
4.3	Bewertung der Modelgüte . . . . .	29
4.4	Erkenntnisse . . . . .	33
4.5	Beste Modelle . . . . .	38
4.6	Fazit . . . . .	42
<b>5</b>	<b>ARIMA</b>	<b>43</b>
5.1	Theretische Überlegungen . . . . .	43
5.2	Stationarität (Bedingungen erfüllt?) . . . . .	43
5.3	Praxis . . . . .	47
5.4	Bewertung der Modelgüte . . . . .	51
5.5	Erkenntnisse . . . . .	59

5.6	Beste Modelle . . . . .	59
5.7	Fazit . . . . .	63
6	Modelle im Vergleich	64
7	Future Reserch	65
	Technical Appendix	66

# 1 Einführung

## 1.1 Wofür?

Vorhersage für Verkehr sinnvoll, für effizientes Verkehrsmanagement: - Vermeidung von Staus (Sicherheits- und Umweltaspekt) - Bessere Verkehrsplanung (z.B. für Buspläne) - Optimierung von Umleitungen/Ampelschaltungen - Navigation

## 1.2 Was?

Vorliegende Daten: - Daten von Bundesanstalt für Straßenwesen - Zeitraum: 1 Jahr (2022) - Aggregation: stündlich - Aufteilung: liegt in 4 Ebenen vor (Kfz (1), Lkw-ähnlich und Pkw-ähnlich (2), ... (5+1),... (8+1))

# 2 Daten

## 2.1 Laden und Aufbereitung der Daten

```
1 df <- read.csv("zst5119_2022.csv", sep = ";")
2 df$Datum <- as.character(df$Datum)
3
4 df <- df %>%
5   mutate(
6     Datetime = as.POSIXct(
7       paste0(
8         "20", substr(Datum, 1, 2), "-",
9         substr(Datum, 3, 4), "-",
10        substr(Datum, 5, 6), " ",
11        sprintf("%02d", Stunde), ":00"
12      ),
13      format = "%Y-%m-%d %H:%M#",
14      #tz = "CET"
15    )
16  )
```

### 2.1.1 Erste Beobachtungen

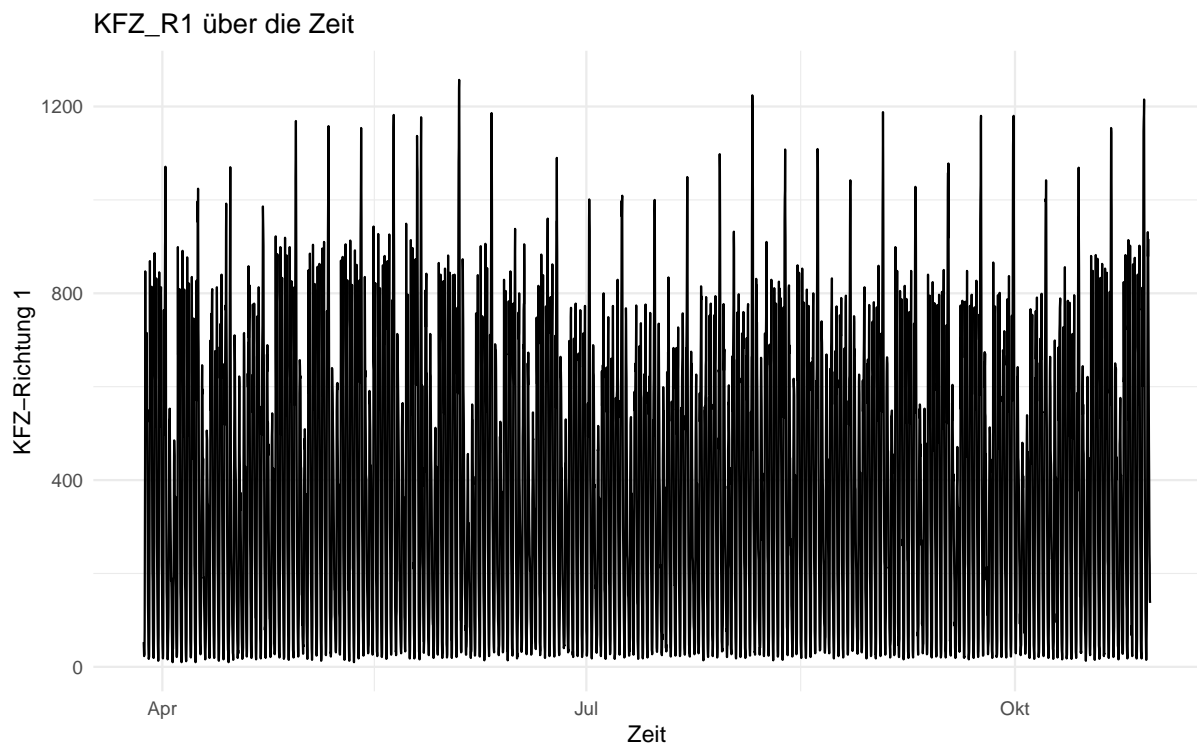
Sommerzeit

```
1 df_summer <- df %>%
2   filter(Datetime >= as.POSIXct("2022-03-28 00:00:00"), Datetime <= as.POSIXct("2022-10-31 00:00:00"))
3
4 traffic_summer_ts <- df_summer %>%
5   select(Datetime, Stunde, Wotag, Fahrtzw, KFZ_R1, KFZ_R2) %>%
6   as_tsibble(index = Datetime)
```

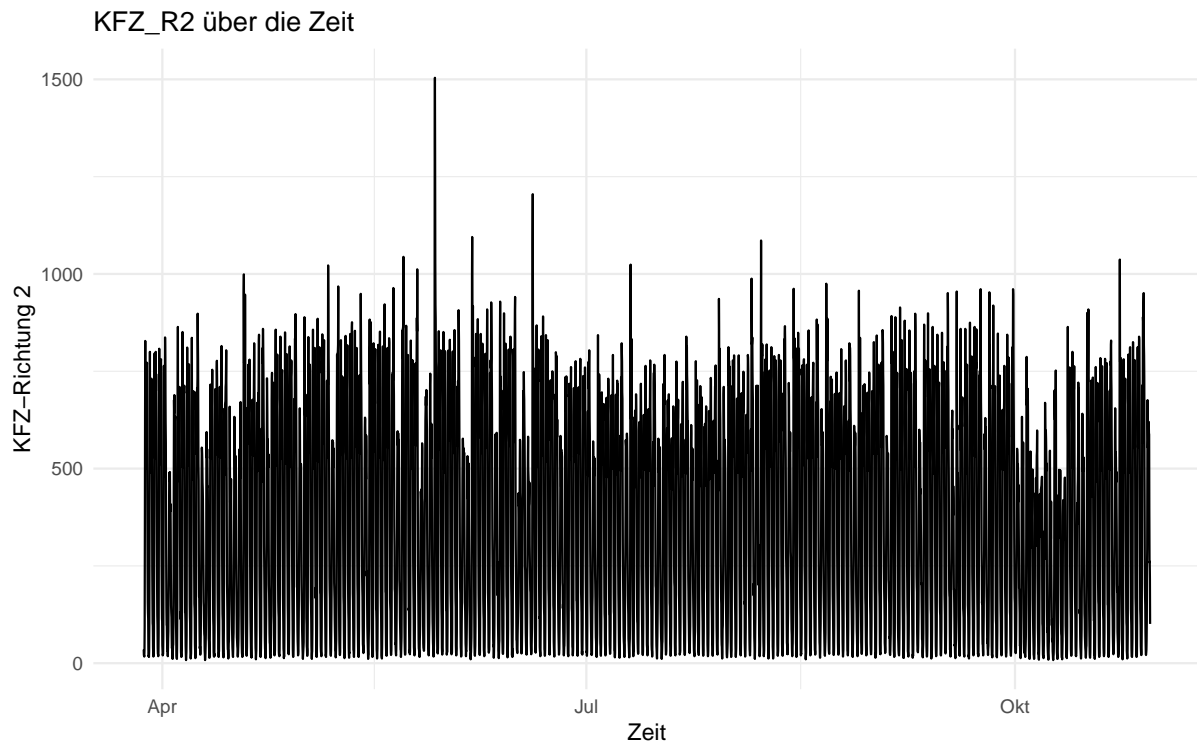
```
traffic_summer_ts
```

```
## # A tsibble: 5,185 x 6 [1h] <?>
##   Datetime           Stunde Wotag Fahrtzw KFZ_R1 KFZ_R2
##   <dtm>             <int> <int> <chr>   <int> <int>
## 1 2022-03-28 00:00:00     24     7 s       53    35
## 2 2022-03-28 01:00:00      1     1 w       39    20
## 3 2022-03-28 02:00:00      2     1 w       23    20
## 4 2022-03-28 03:00:00      3     1 w       26    17
## 5 2022-03-28 04:00:00      4     1 w       27    33
## 6 2022-03-28 05:00:00      5     1 w       78    95
## 7 2022-03-28 06:00:00      6     1 w      277   349
## 8 2022-03-28 07:00:00      7     1 w     439   638
## 9 2022-03-28 08:00:00      8     1 w     847   828
## 10 2022-03-28 09:00:00     9     1 w     562   522
## # i 5,175 more rows
```

```
autoplot(traffic_summer_ts, KFZ_R1) +
  labs(title = "KFZ_R1 über die Zeit", x = "Zeit", y = "KFZ-Richtung 1") +
  theme_minimal()
```



```
autoplot(traffic_summer_ts, KFZ_R2) +
  labs(title = "KFZ_R2 über die Zeit", x = "Zeit", y = "KFZ-Richtung 2") +
  theme_minimal()
```

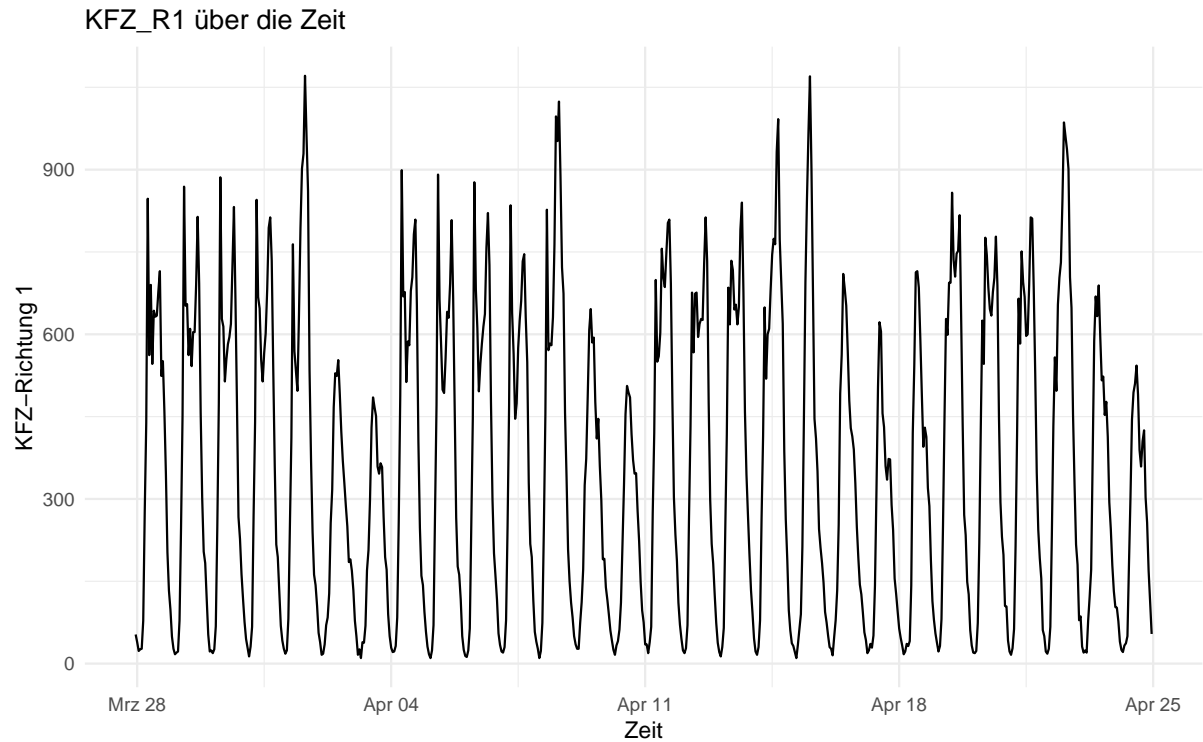


Ersten 4 Wochen aus Sommerzeit

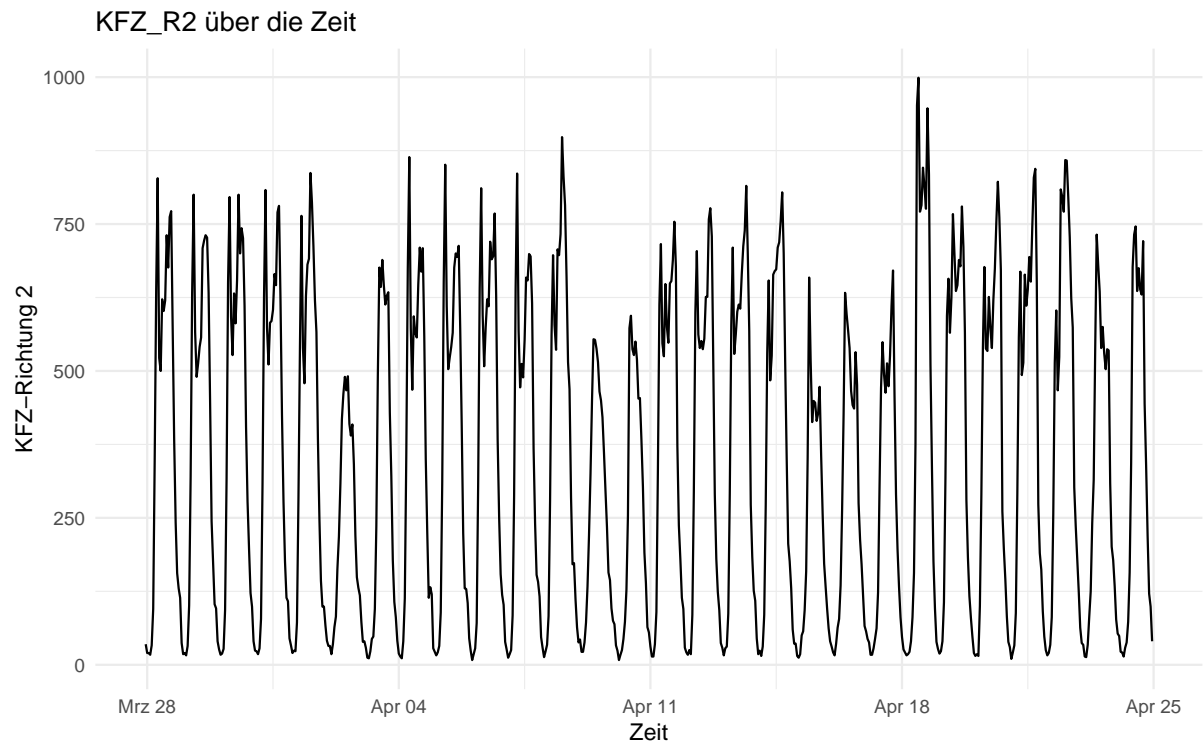
```
1 df_4weeks <- df %>%
2   filter(Datetime >= as.POSIXct("2022-03-28 00:00:00"), Datetime <= as.POSIXct("2022-04-28 00:00:00"))
3
4 traffic_4weeks_ts <- df_4weeks %>%
5   select(Datetime, Stunde, Wotag, Fahrtzw, KFZ_R1, KFZ_R2) %>%
6   as_tsibble(index = Datetime)
7
8 traffic_4weeks_ts
```

```
## # A tsibble: 673 x 6 [1h] <?>
##   Datetime           Stunde Wotag Fahrtzw KFZ_R1 KFZ_R2
##   <dtm>             <int> <int> <chr>   <int> <int>
## 1 2022-03-28 00:00:00     24     7 s       53    35
## 2 2022-03-28 01:00:00      1     1 w       39    20
## 3 2022-03-28 02:00:00      2     1 w       23    20
## 4 2022-03-28 03:00:00      3     1 w       26    17
## 5 2022-03-28 04:00:00      4     1 w       27    33
## 6 2022-03-28 05:00:00      5     1 w       78    95
## 7 2022-03-28 06:00:00      6     1 w      277   349
## 8 2022-03-28 07:00:00      7     1 w     439   638
## 9 2022-03-28 08:00:00      8     1 w     847   828
##10 2022-03-28 09:00:00      9     1 w     562   522
## # i 663 more rows
```

```
1 autoplot(traffic_4weeks_ts, KFZ_R1) +
2   labs(title = "KFZ_R1 über die Zeit", x = "Zeit", y = "KFZ-Richtung 1") +
3   theme_minimal()
```



```
1 autoplot(traffic_4weeks_ts, KFZ_R2) +  
2   labs(title = "KFZ_R2 über die Zeit", x = "Zeit", y = "KFZ-Richtung 2") +  
3   theme_minimal()
```



```
1 # Ostern 17.4.22
```

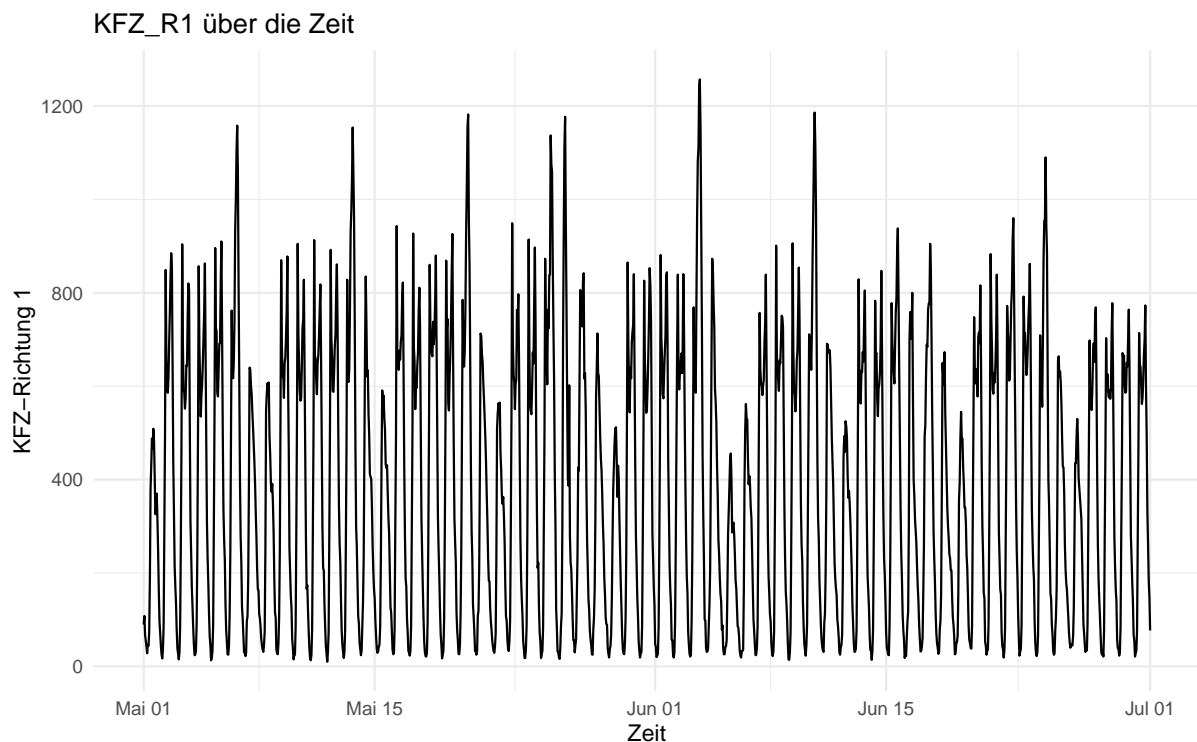
Zweiter und dritter Monate aus Sommerzeit



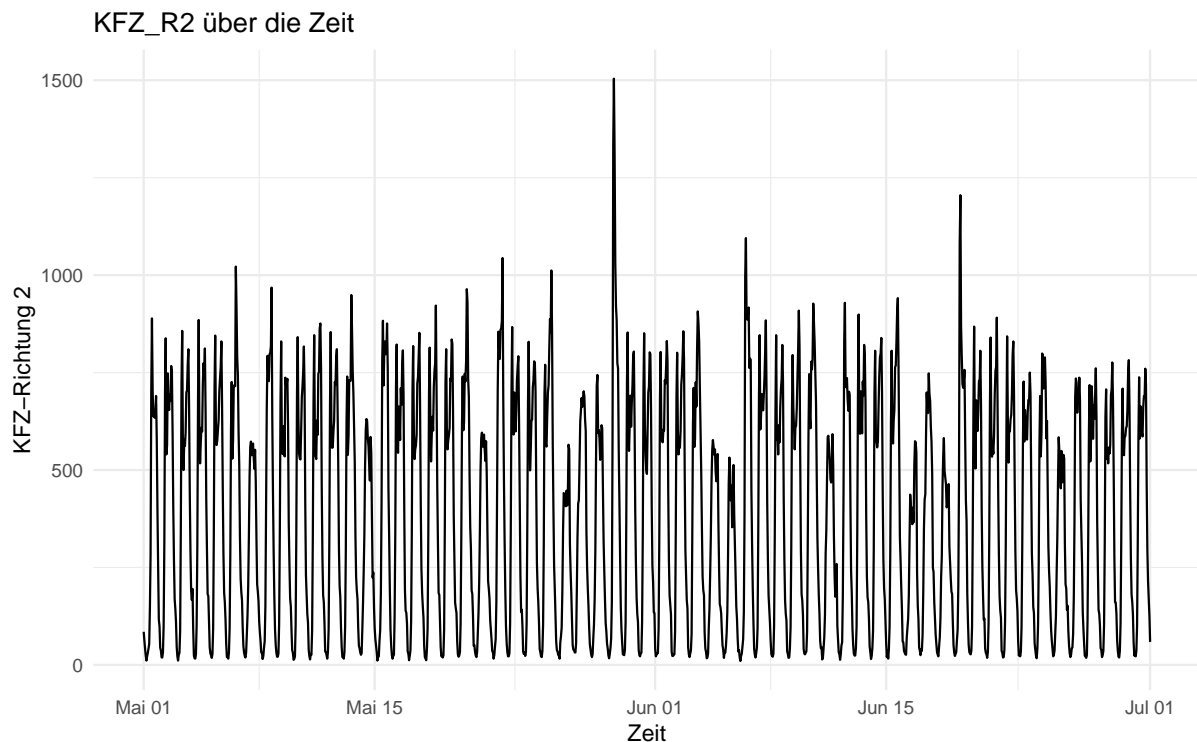
```
1 df_2months <- df %>%
2   filter(Datetime >= as.POSIXct("2022-05-01 00:00:00"), Datetime <= as.POSIXct("2022-07-01 00:00:00"))
3
4 traffic_2months_ts <- df_2months %>%
5   select(Datetime, Stunde, Wotag, Fahrtzw, KFZ_R1, KFZ_R2) %>%
6   as_tsibble(index = Datetime)
7
8 traffic_2months_ts
```

```
## # A tsibble: 1,465 x 6 [1h] <?>
##   Datetime           Stunde Wotag Fahrtzw KFZ_R1 KFZ_R2
##   <dtm>              <int> <int> <chr>   <int> <int>
## 1 2022-05-01 00:00:00      24     6 w        90     85
## 2 2022-05-01 01:00:00       1     7 s       108     62
## 3 2022-05-01 02:00:00       2     7 s        67     41
## 4 2022-05-01 03:00:00       3     7 s        55     23
## 5 2022-05-01 04:00:00       4     7 s        42     11
## 6 2022-05-01 05:00:00       5     7 s        28     22
## 7 2022-05-01 06:00:00       6     7 s        47     31
## 8 2022-05-01 07:00:00       7     7 s        44     42
## 9 2022-05-01 08:00:00       8     7 s        74     54
## 10 2022-05-01 09:00:00      9     7 s       190    146
## # i 1,455 more rows
```

```
1 autoplot(traffic_2months_ts, KFZ_R1) +
2   labs(title = "KFZ_R1 über die Zeit", x = "Zeit", y = "KFZ-Richtung 1") +
3   theme_minimal()
```



```
1 autoplot(traffic_2months_ts, KFZ_R2) +  
2   labs(title = "KFZ_R2 über die Zeit", x = "Zeit", y = "KFZ-Richtung 2") +  
3   theme_minimal()
```



```
1 # Christi Himmelfahrt 26.05.22  
2 # Pfingsten 05.06.22 (aber So)  
3 # Beginn Sommerferien NRW 27.06.2022
```

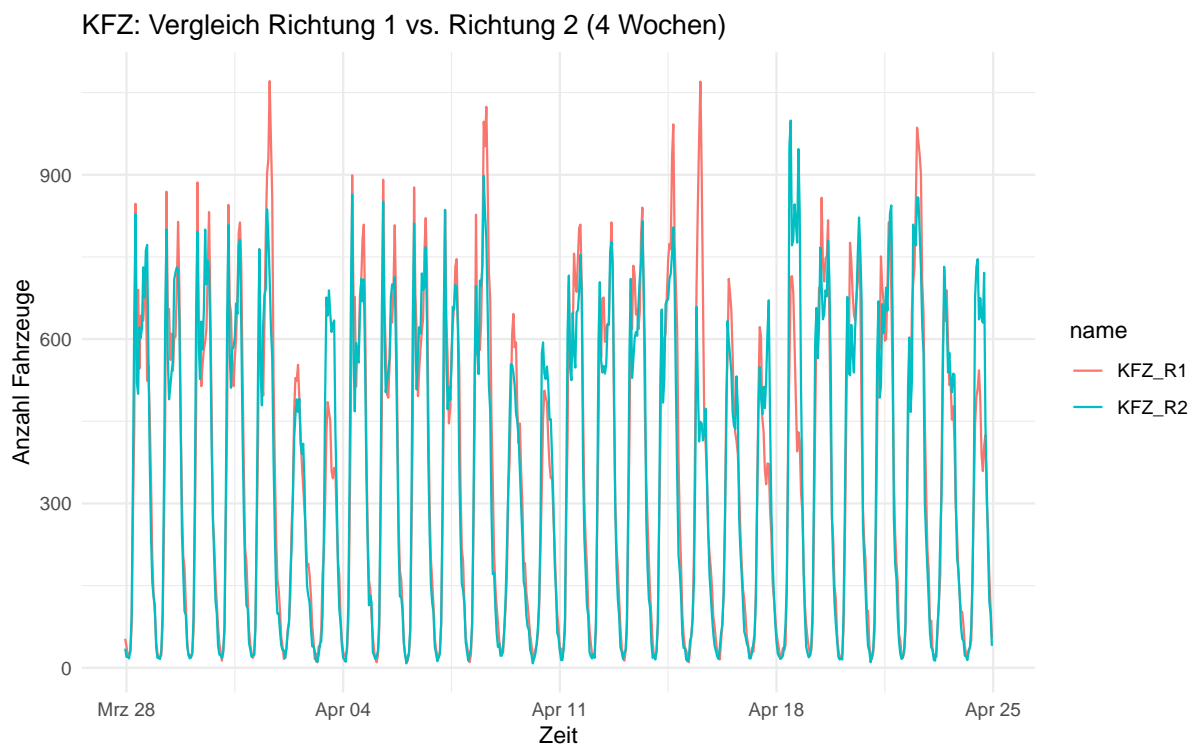
### 2.1.2 Erste Erkenntnisse

- Wochensaison erkennbar: Ausschläge Mo-Do etwa gleich stark, Fr am stärksten, Sa + So deutlich schwächer ausgeprägt (Sa nochmal mehr als So)
- Mo - Fr Mittagstief erkennbar
- Auffälligkeiten, wie höheres/niedriges Verkehrsaufkommen, Zusammenhang mit Feiertagen/Ferien (Mo 27.05. -> Christi Himmelfahrt)
- Probleme durch Zeitverschiebung -> Fokus auf Stunden oder Tage bzw. tages- oder wochensaison?

Stunden oder Tage? - Stunden: - Tagessaison erkennbar (Mittagstief) - Nachteil: Zeitverschiebung -> Mittelwerte bilden. Einmal aus Stunde vorher/stunde nachher, einmal aus den beiden Zeilen (gleiche Uhrzeit) - Tage: - wahrscheinlich Komplexere/interessantere Struktur über die Zeit, als bei Stunden - Wochensaison - besser für Model

Richtungen einzeln oder zusammen? - Einzeln - mehr Details/feinere Muster (Berufsverkehr, Ferien etc.) -> Aufgabenstellung?(sinnvoll bei Beobachtung von Berufsverkehr, Navigation/Stau) - aber mehr Arbeit/Plots -> schwierigerer Überblick - einfacher zu modellieren - Zusammen - Gesamtüberblick besser -> Aufgabenstellung?(sinnvoll bei Lärmbelästigung oder Umweltbelastung) - weniger Details -> ungünstig für ETS, ARIMA (stabile Muster werden weggelätet) - mehr Arbeit bei Modellierung/komplexer, aber realistischer

```
1 traffic_4weeks_long <- traffic_4weeks_ts %>%
2   pivot_longer(cols = c(KFZ_R1, KFZ_R2),
3                 values_to = "Anzahl")
4
5 # Plot: beide Richtungen in einem Plot
6 autoplot(travel_4weeks_long, Anzahl) +
7   labs(title = "KFZ: Vergleich Richtung 1 vs. Richtung 2 (4 Wochen)",
8         x = "Zeit",
9         y = "Anzahl Fahrzeuge") +
10  theme_minimal()
```



- Beobachtungen:
  - Struktur an Wochentagen sehr ähnlich
  - Auffälligkeiten an Wochenenden/Feiertagen (für Prognose wahrscheinlich sinnvoll -> daher getrennte Richtungen)

## 2.2 Datenaufbereitung für Klassen

- Daten als Aggregation auf Tage in die drei Klassen 1,2, und 5+1
  - 1: KFZ
  - 2: PKW-ähnliche und LKW-ähnliche
  - 3: Aufteilung der Klassen in (5+1): nicht-klassifizierbare Kfz, Pkw-Gruppe, Pkw mit Anhänger, Lkw > 3,5t o. Anhänger, Lkw > 3,5t m. Anhänger / Sattelkraftfahrzeuge, Busse

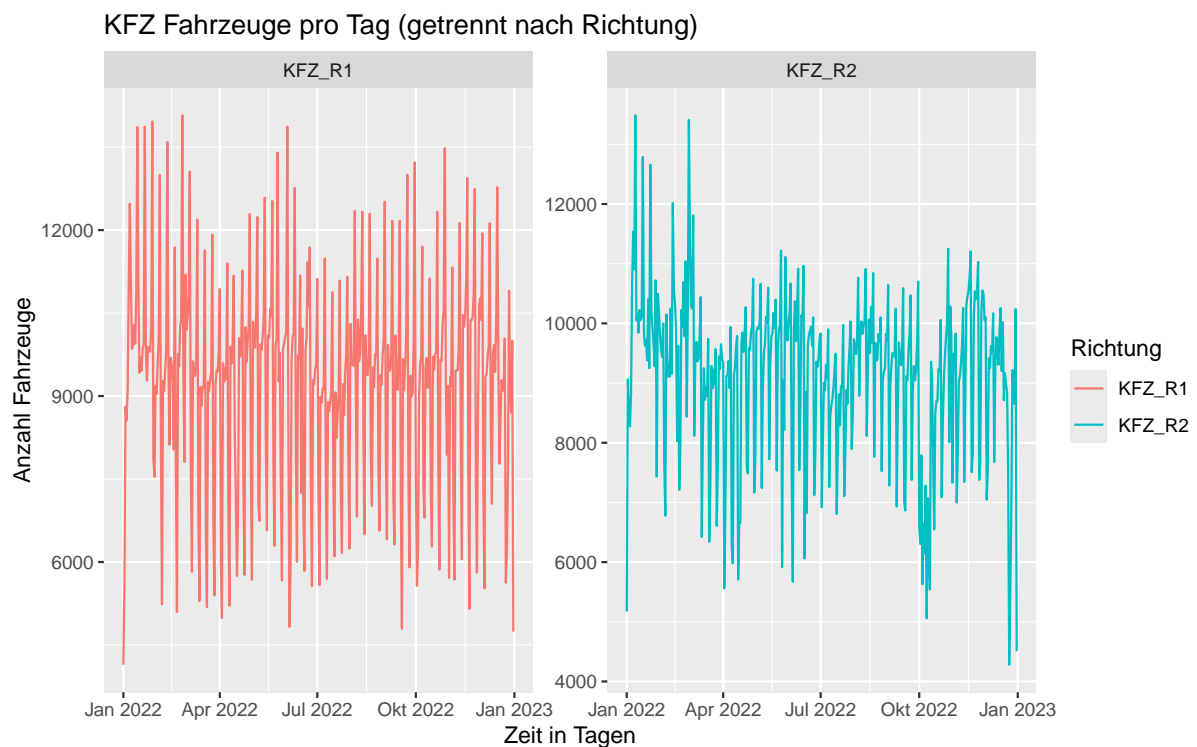
```
1 df <- df %>%
2   mutate(Datum_tag = as.Date(Datetime)) %>%
3   filter(!is.na(Datum_tag))
```

```
4
5 # Aggregation nach Tagen
6
7 # KFZ (beide Richtung)
8 traffic_kfz_ts <- df %>%
9   group_by(Datum_tag) %>%
10  summarise(
11    KFZ_R1 = sum(KFZ_R1, na.rm = TRUE),
12    KFZ_R2 = sum(KFZ_R2, na.rm = TRUE)
13  ) %>%
14  ungroup() %>%
15  pivot_longer(cols = c(KFZ_R1, KFZ_R2),
16               names_to = "Richtung",
17               values_to = "Anzahl") %>%
18  as_tsibble(index = Datum_tag, key = Richtung)
19
20 # Pkw/Lkw ähnlich (beide Richtung)
21 traffic_pkw_lkw_ts <- df %>%
22   group_by(Datum_tag) %>%
23   summarise(
24     Pkw_R1 = sum(PLZ_R1, na.rm = TRUE),
25     Pkw_R2 = sum(PLZ_R2, na.rm = TRUE),
26     Lkw_R1 = sum(Lkw_R1, na.rm = TRUE),
27     Lkw_R2 = sum(Lkw_R2, na.rm = TRUE)
28   ) %>%
29   ungroup() %>%
30   pivot_longer(cols = -Datum_tag,
31                names_to = "Kategorie",
32                values_to = "Anzahl") %>%
33   as_tsibble(index = Datum_tag, key = Kategorie)
34
35 # 5+1 (beide Richtung)
36 traffic_pkw_lkw_bus_ts1 <- df %>%
37   group_by(Datum_tag) %>%
38   summarise(
39     Nicht_klassifizierbar_R1 = sum(Son_R1, na.rm = TRUE),
40     Nicht_klassifizierbar_R2 = sum(Son_R2, na.rm = TRUE),
41     Pkw_Gruppe_R1 = sum(Pkw_R1, Lfw_R1, Mot_R1, na.rm = TRUE),
42     Pkw_Gruppe_R2 = sum(Pkw_R2, Lfw_R2, Mot_R2, na.rm = TRUE),
43     Pkw_mit_Anh_R1 = sum(PmA_R1, na.rm = TRUE),
44     Pkw_mit_Anh_R2 = sum(PmA_R2, na.rm = TRUE),
45     Lkw_ohne_Anh_R1 = sum(LoA_R1, na.rm = TRUE),
46     Lkw_ohne_Anh_R2 = sum(LoA_R2, na.rm = TRUE),
47     Lkw_mA_R1 = sum(Lzg_R1, Sat_R1, na.rm = TRUE),
48     Lkw_mA_R2 = sum(Lzg_R2, Sat_R2, na.rm = TRUE),
49     Busse_R1 = sum(Bus_R1, na.rm = TRUE),
50     Busse_R2 = sum(Bus_R2, na.rm = TRUE)
51   ) %>%
52   ungroup() %>%
53   pivot_longer(cols = -Datum_tag,
54                names_to = "Kategorie",
```

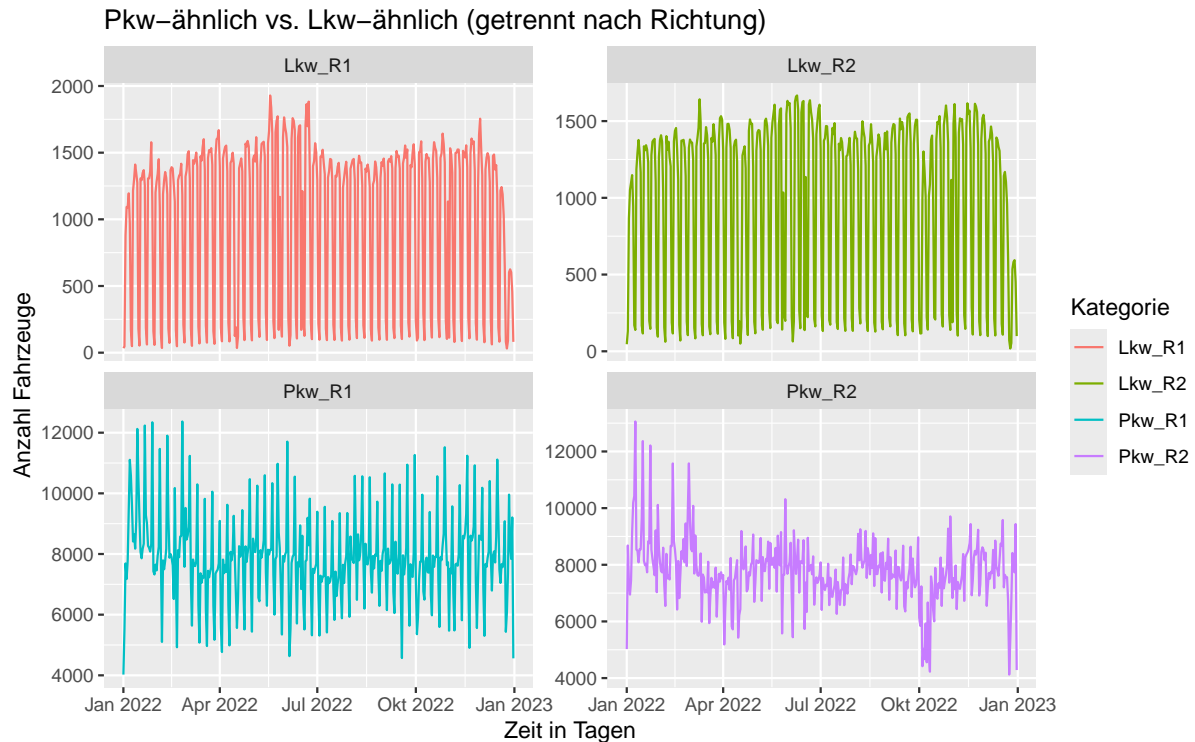
```
55 values_to = "Anzahl") %>%
56 as_tsibble(index = Datum_tag, key = Kategorie)
```

- Alle einmal plotten, um zu sehen was für forecast lohnt (genug Fahrzeuge)

```
1 # KFZ (beide Richtung)
2 autoplot(traffic_kfz_ts, Anzahl) +
3   facet_wrap(~Richtung, scales = "free_y") +
4   labs(title = "KFZ Fahrzeuge pro Tag (getrennt nach Richtung)",
5         y = "Anzahl Fahrzeuge", x = "Zeit in Tagen")
```



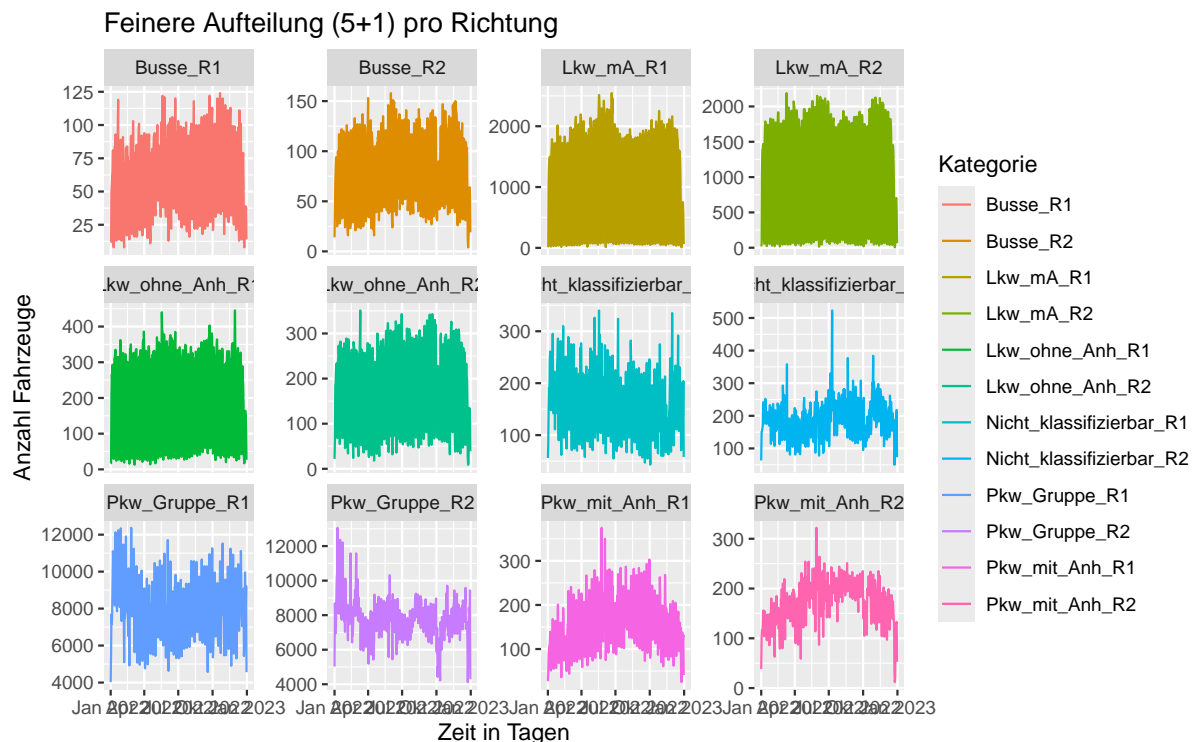
```
1 # Pkw/Lkw ähnlich (beide Richtung)
2 autoplot(traffic_pkw_lkw_ts, Anzahl) +
3   facet_wrap(~Kategorie, scales = "free_y") +
4   labs(title = "Pkw-ähnlich vs. Lkw-ähnlich (getrennt nach Richtung)",
5         y = "Anzahl Fahrzeuge", x = "Zeit in Tagen")
```



```

1 # 5+1 (beide Richtung)
2 autoplot(traffic_pkw_lkw_bus_ts1, Anzahl) +
3   facet_wrap(~Kategorie, scales = "free_y") +
4   labs(title = "Feinere Aufteilung (5+1) pro Richtung",
5         y = "Anzahl Fahrzeuge", x = "Zeit in Tagen")

```



## 2.3 Beobachtungen aus 5+1 auf Tagesebene:

- Richtung 1 schwankt bei allen Klassen deutlich stärker (größere Varianz) als Richtung 2
- überall Wochensaison erkennbar
- PKW
  - mit Abstand das größte Aufkommen (ca. 8000/Tag) -> dadurch auch größte Streuung bei R1
  - trotzdem stabile Struktur
- LKW (oA)
  - kaum Schwankungen erkennbar (ca. 250/Tag)
  - geringes Aufkommen, aber für Wirtschaft/Struktur evtl. interessant
- LKW (mA)
  - geringe Schwankungen
  - höheres Aufkommen als LKW (oA) -> ca. 1500/Tag
- Busse
  - geringe Schwankungen
  - evtl. für ÖPNV interessant
- PKW (mA)
  - auffällige Saisonalität erkennbar -> Sommer mehr als Winter
  - ca. 150/Tag
- nicht klassifizierbare
  - ein deutlich größerer Wert bei R2 erkennbar
  - ca. 200/Tag

## 2.4 Überlegungen für weiteres Vorgehen

### 2.4.1 Worauf Fokus? Welche Klassen sinnvoll (Zahlen, Hintergrund)?

- PKW
  - generell relevant
  - mit Anhänger, geringes Aufkommen, viel Rauschen
- LKW mit Anhänger
  - ohne Anhänger ähnliche Struktur, aber weniger Aufkommen, deswegen Entscheidung für LKW mit Anhänger
  - wirtschaftlich interessant
- Busse
  - zwar geringe Zahl, aber nochmal andere Struktur als PKW und LKW
  - für ÖPNV interessant Nicht klassifizierbare Fahrzeuge werden ebenfalls nicht weiter betrachtet. Eine Prognose wäre hier aufgrund der verschiedenen Fahrzeuge nicht sinnvoll.

```
1 # 5+1 (beide Richtung)
2 traffic_pkw_lkw_bus_ts <- df %>%
3   group_by(Datum_tag) %>%
4   summarise(
5     #Nicht_klassifizierbar_R1 = sum(Son_R1, na.rm = TRUE),
6     #Nicht_klassifizierbar_R2 = sum(Son_R2, na.rm = TRUE),
7     Pkw_Gruppe_R1 = sum(Pkw_R1, Lfw_R1, Mot_R1, na.rm = TRUE),
```

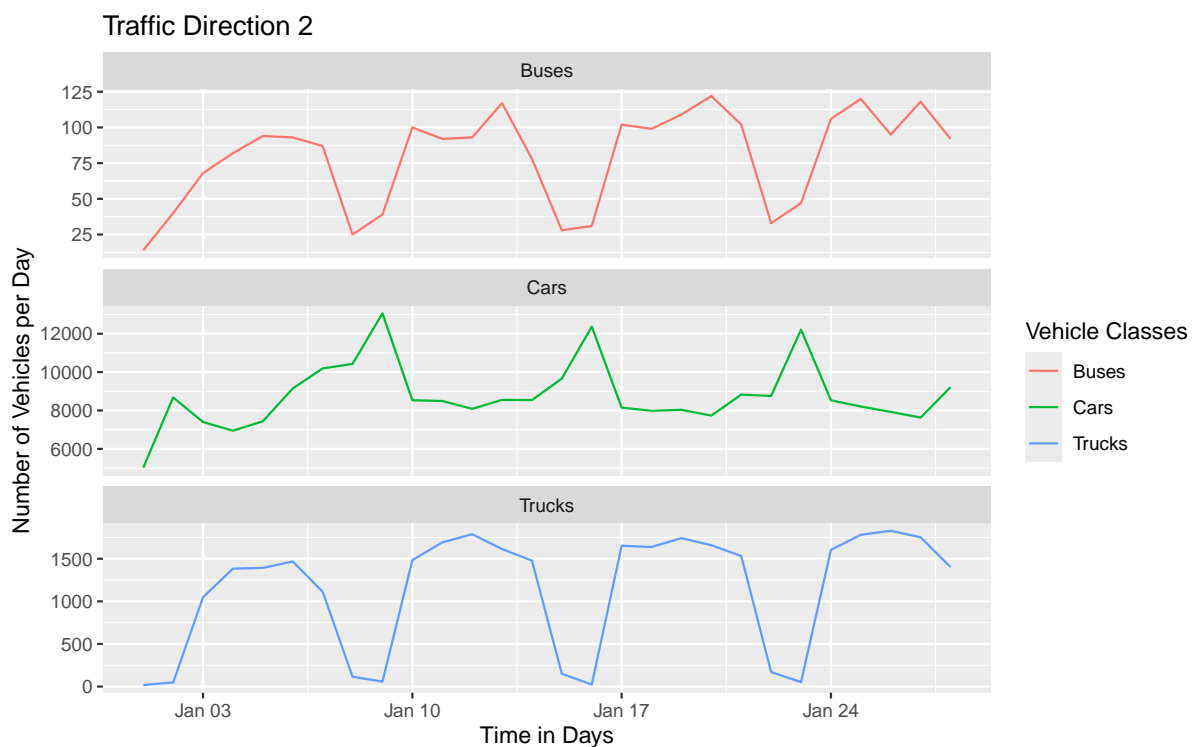
```
8   Pkw_Gruppe_R2 = sum(Pkw_R2, Lfw_R2, Mot_R2, na.rm = TRUE),
9   #Pkw_mit_Anh_R1 = sum(PmA_R1, na.rm = TRUE),
10  #Pkw_mit_Anh_R2 = sum(PmA_R2, na.rm = TRUE),
11  #Lkw_ohne_Anh_R1 = sum(LoA_R1, na.rm = TRUE),
12  #Lkw_ohne_Anh_R2 = sum(LoA_R2, na.rm = TRUE),
13  Lkw_mA_R1 = sum(Lzg_R1, Sat_R1, na.rm = TRUE),
14  Lkw_mA_R2 = sum(Lzg_R2, Sat_R2, na.rm = TRUE),
15  Busse_R1 = sum(Bus_R1, na.rm = TRUE),
16  Busse_R2 = sum(Bus_R2, na.rm = TRUE)
17 ) %>%
18 ungroup() %>%
19 as_tsibble(index = Datum_tag)
```

```
1  # Pkw
2  pkw_R1_ts <- traffic_pkw_lkw_bus_ts %>%
3    select(Datum_tag, Pkw_Gruppe_R1) %>%
4    as_tsibble(index = Datum_tag)
5
6  pkw_R2_ts <- traffic_pkw_lkw_bus_ts %>%
7    select(Datum_tag, Pkw_Gruppe_R2) %>%
8    as_tsibble(index = Datum_tag)
9
10 # Lkw
11 lkw_R1_ts <- traffic_pkw_lkw_bus_ts %>%
12   select(Datum_tag, Lkw_mA_R1) %>%
13   as_tsibble(index = Datum_tag)
14
15 lkw_R2_ts <- traffic_pkw_lkw_bus_ts %>%
16   select(Datum_tag, Lkw_mA_R2) %>%
17   as_tsibble(index = Datum_tag)
18
19 # Busse
20 bus_R1_ts <- traffic_pkw_lkw_bus_ts %>%
21   select(Datum_tag, Busse_R1) %>%
22   as_tsibble(index = Datum_tag)
23
24 bus_R2_ts <- traffic_pkw_lkw_bus_ts %>%
25   select(Datum_tag, Busse_R2) %>%
26   as_tsibble(index = Datum_tag)
```

```
1  traffic_r2_long <- traffic_pkw_lkw_bus_ts %>%
2    select(Datum_tag, Pkw_Gruppe_R2, Lkw_mA_R2, Busse_R2) %>%
3    pivot_longer(
4      cols = c(Pkw_Gruppe_R2, Lkw_mA_R2, Busse_R2),
5      names_to = "Fahrzeugklasse",
6      values_to = "Anzahl"
7    ) %>%
8    mutate(
9      Fahrzeugklasse = recode(Fahrzeugklasse,
```



```
10   "Pkw_Gruppe_R2" = "Cars",
11   "Lkw_mA_R2" = "Trucks",
12   "Busse_R2" = "Buses"
13 )
14 ) %>%
15 as_tsibble(index = Datum_tag, key = Fahrzeugklasse)
16
17 # Ersten 28 Tage
18 traffic_r2_4w <- traffic_r2_long %>%
19   filter(Datum_tag <= min(Datum_tag) + 27)
20
21 # Plot
22 autoplot(traffic_r2_4w, Anzahl) +
23   facet_wrap(~ Fahrzeugklasse, scales = "free_y", ncol = 1) +
24   labs(
25     title = "Traffic Direction 2",
26     x = "Time in Days",
27     y = "Number of Vehicles per Day"
28   ) +
29   guides(color = guide_legend(title = "Vehicle Classes"))
```



### 2.4.2 Prognosezeitraum

Die Daten sind jetzt auf Tagesebene (also 365 Datenpunkte). Es liegt eine Wochensaison vor, die sich 51x wiederholt. Erlaubt robuste Modellschätzung. Für Prognosezeitraum werden vier Wochen gewählt, um zu bewerten, ob die Wochensaison gut prognostiziert wird. Länger Zeitraum wäre aufgrund mehrerer Faktoren schwierig (Karneval, Ostern...). (Wie Feiertage an Modell übergeben)

## 3 Benchmarks

### 3.1 Trainings-/Testsplit

- 80/20 Trainings-/Teststplit

```
1 h <- 35
2
3 end_date <- max(traffic_kfz_ts$Datum_tag)
4 start_test <- end_date - h + 1
5
6 # PKW/LKW/Bus (5+1)
7 #train_pkw_lkw_bus <- traffic_pkw_lkw_bus_ts %>% filter(Datum_tag < start_test)
8 #test_pkw_lkw_bus <- traffic_pkw_lkw_bus_ts %>% filter(Datum_tag >= start_test)
9
10
11 # PKW
12 train_pkw_R1 <- pkw_R1_ts %>% filter(Datum_tag < start_test)
13 test_pkw_R1 <- pkw_R1_ts %>% filter(Datum_tag >= start_test)
14
15 train_pkw_R2 <- pkw_R2_ts %>% filter(Datum_tag < start_test)
16 test_pkw_R2 <- pkw_R2_ts %>% filter(Datum_tag >= start_test)
17
18 # LKW
19 train_lkw_R1 <- lkw_R1_ts %>% filter(Datum_tag < start_test)
20 test_lkw_R1 <- lkw_R1_ts %>% filter(Datum_tag >= start_test)
21
22 train_lkw_R2 <- lkw_R2_ts %>% filter(Datum_tag < start_test)
23 test_lkw_R2 <- lkw_R2_ts %>% filter(Datum_tag >= start_test)
24
25 # Busse
26 train_bus_R1 <- bus_R1_ts %>% filter(Datum_tag < start_test)
27 test_bus_R1 <- bus_R1_ts %>% filter(Datum_tag >= start_test)
28
29 train_bus_R2 <- bus_R2_ts %>% filter(Datum_tag < start_test)
30 test_bus_R2 <- bus_R2_ts %>% filter(Datum_tag >= start_test)
```

### 3.2 Naive Methoden

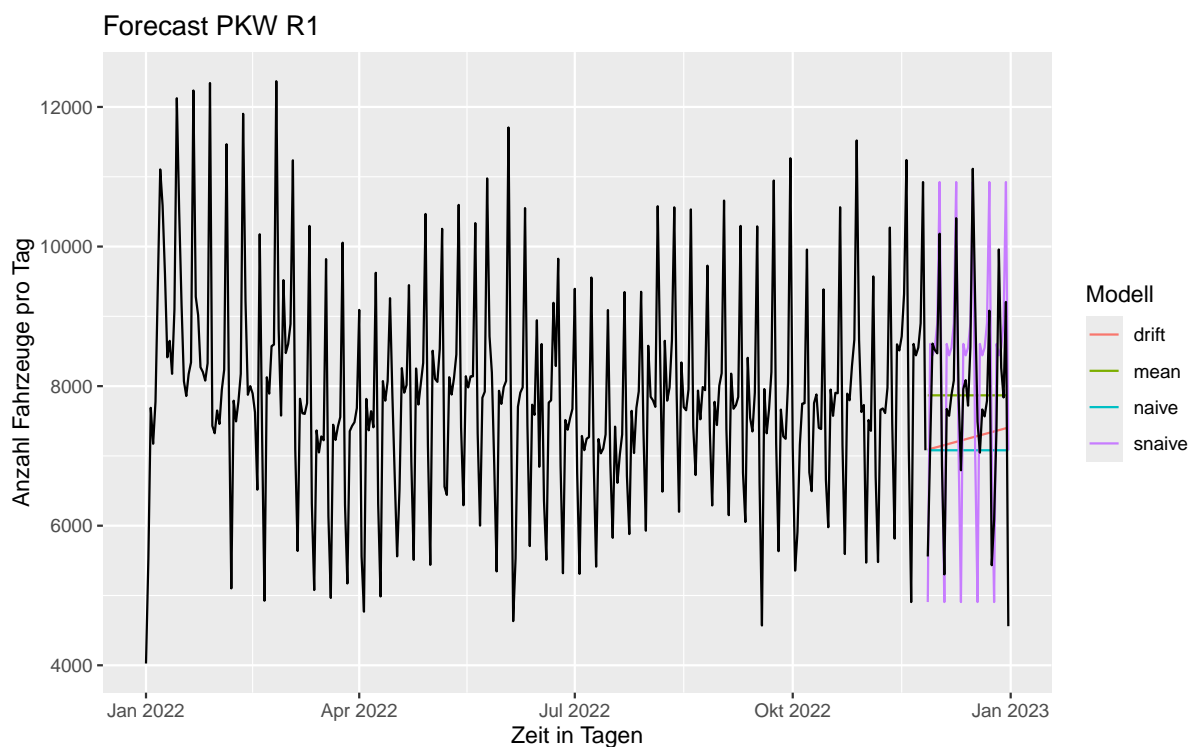
```
1 #models_pkw_lkw_bus <- train_pkw_lkw_bus %>%
2   #model(
3     #mean    = MEAN(Pkw_Gruppe_R1),
4     #naive    = NAIVE(Pkw_Gruppe_R1),
5     #snaive   = SNAIVE(Pkw_Gruppe_R1 ~ period("7 days")),
6     #drift    = RW(Pkw_Gruppe_R1 ~ drift())
7   #)
8
9 #fc_pkw_lkw_bus <- models_pkw_lkw_bus %>% forecast(h = h)
10
11 # PKW
```

```
12 models_pkw_R1 <- train_pkw_R1 %>%
13   model(
14     mean    = MEAN(Pkw_Gruppe_R1),
15     naive   = NAIVE(Pkw_Gruppe_R1),
16     snaive  = SNAIVE(Pkw_Gruppe_R1),
17     drift   = RW(Pkw_Gruppe_R1 ~ drift())
18   )
19 fc_pkw_R1 <- models_pkw_R1 %>% forecast(h = h)
20
21 models_pkw_R2 <- train_pkw_R2 %>%
22   model(
23     mean    = MEAN(Pkw_Gruppe_R2),
24     naive   = NAIVE(Pkw_Gruppe_R2),
25     snaive  = SNAIVE(Pkw_Gruppe_R2),
26     drift   = RW(Pkw_Gruppe_R2 ~ drift())
27   )
28 fc_pkw_R2 <- models_pkw_R2 %>% forecast(h = h)
29
30
31 # LKW
32 models_lkw_R1 <- train_lkw_R1 %>%
33   model(
34     mean    = MEAN(Lkw_mA_R1),
35     naive   = NAIVE(Lkw_mA_R1),
36     snaive  = SNAIVE(Lkw_mA_R1),
37     drift   = RW(Lkw_mA_R1 ~ drift())
38   )
39 fc_lkw_R1 <- models_lkw_R1 %>% forecast(h = h)
40
41 models_lkw_R2 <- train_lkw_R2 %>%
42   model(
43     mean    = MEAN(Lkw_mA_R2),
44     naive   = NAIVE(Lkw_mA_R2),
45     snaive  = SNAIVE(Lkw_mA_R2),
46     drift   = RW(Lkw_mA_R2 ~ drift())
47   )
48 fc_lkw_R2 <- models_lkw_R2 %>% forecast(h = h)
49
50
51 # Bus
52 models_bus_R1 <- train_bus_R1 %>%
53   model(
54     mean    = MEAN(Busse_R1),
55     naive   = NAIVE(Busse_R1),
56     snaive  = SNAIVE(Busse_R1),
57     drift   = RW(Busse_R1 ~ drift())
58   )
59 fc_bus_R1 <- models_bus_R1 %>% forecast(h = h)
60
61 models_bus_R2 <- train_bus_R2 %>%
62   model(
```

```
63 mean = MEAN(Busse_R2),
64 naive = NAIVE(Busse_R2),
65 snaive = SNAIVE(Busse_R2),
66 drift = RW(Busse_R2 ~ drift())
67 )
68 fc_bus_R2 <- models_bus_R2 %>% forecast(h = h)
```

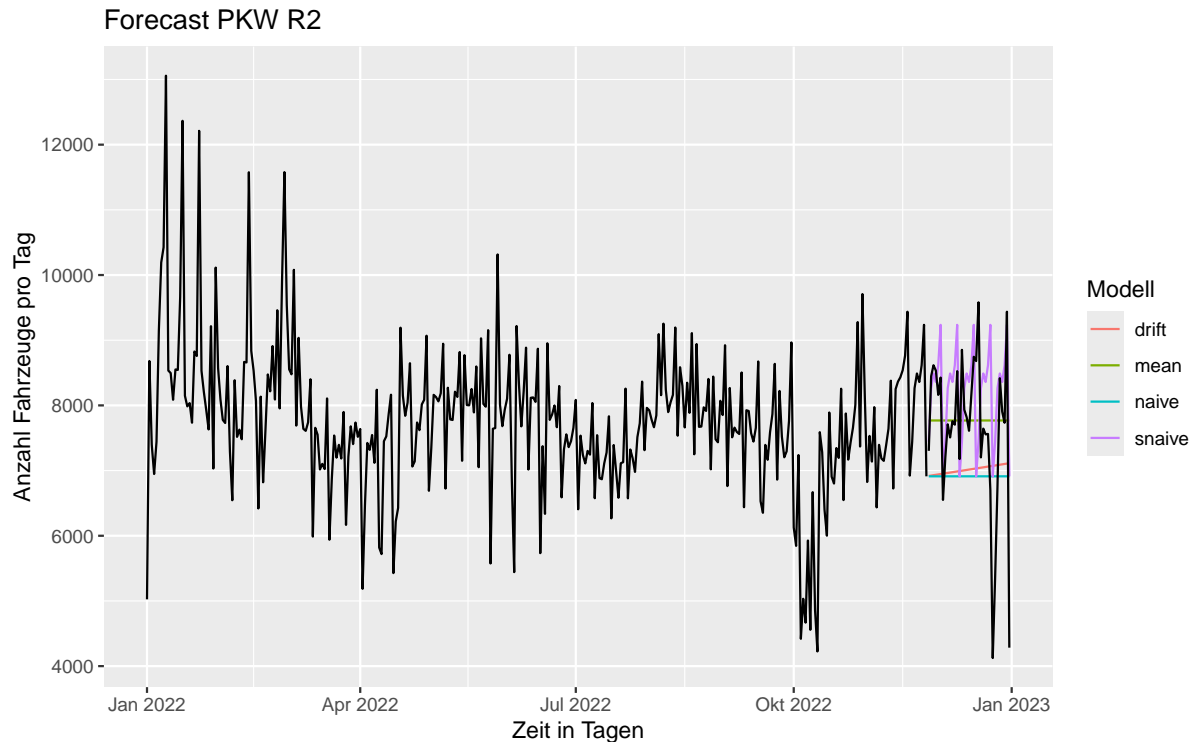
### 3.2.1 PKW/LKW/Bus

```
1 #fc_pkw_lkw_bus %>%
2 #autoplot(train_pkw_lkw_bus, level = NULL) +
3 #autolayer(test_pkw_lkw_bus, Pkw_Gruppe_R1, color = "black") +
4 #facet_wrap(~Kategorie, scales = "free_y") +
5 #labs(title = "Forecasts 5+1-Kategorien", y = "Anzahl Fahrzeuge pro Tag", x = "Datum")
6 #guides(color = guide_legend(title = "Modell"))
7
8 # PKW
9 fc_pkw_R1 %>%
10 autoplot(train_pkw_R1, level = NULL) +
11 autolayer(test_pkw_R1, Pkw_Gruppe_R1, color = "black") +
12 labs(title = "Forecast PKW R1",
13       y = "Anzahl Fahrzeuge pro Tag",
14       x = "Zeit in Tagen") +
15 guides(color = guide_legend(title = "Modell"))
```

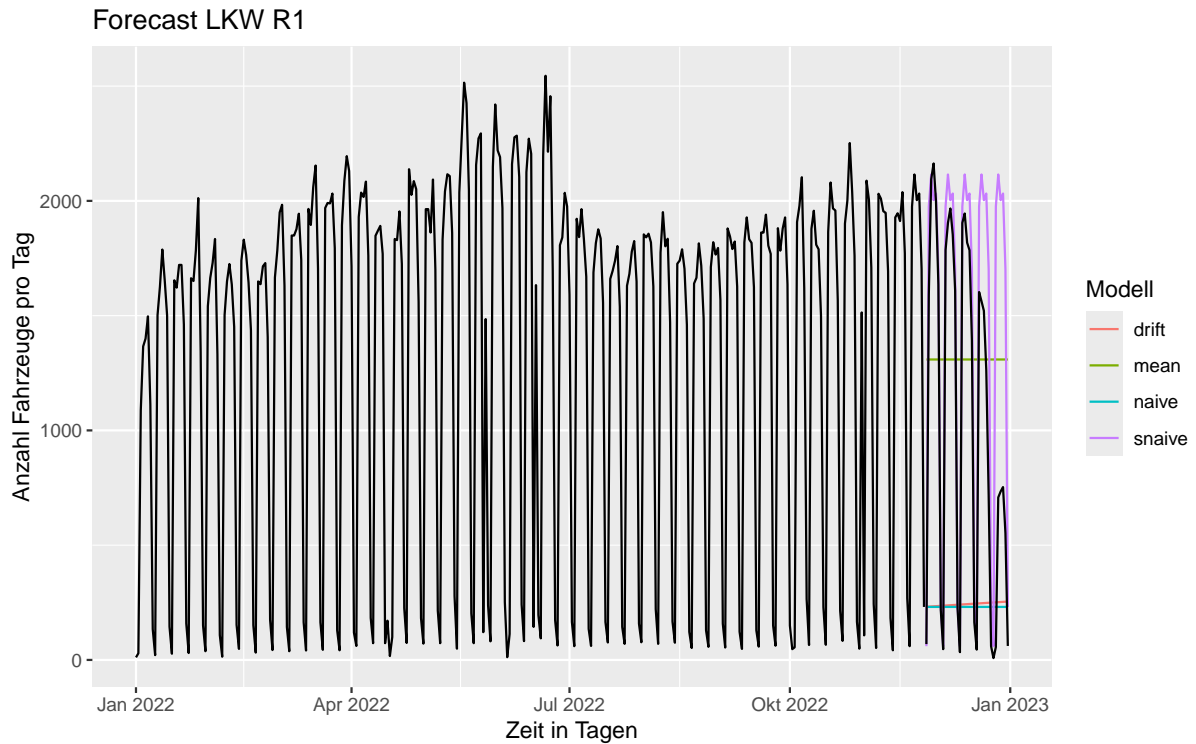


```
1 fc_pkw_R2 %>%
2 autoplot(train_pkw_R2, level = NULL) +
```

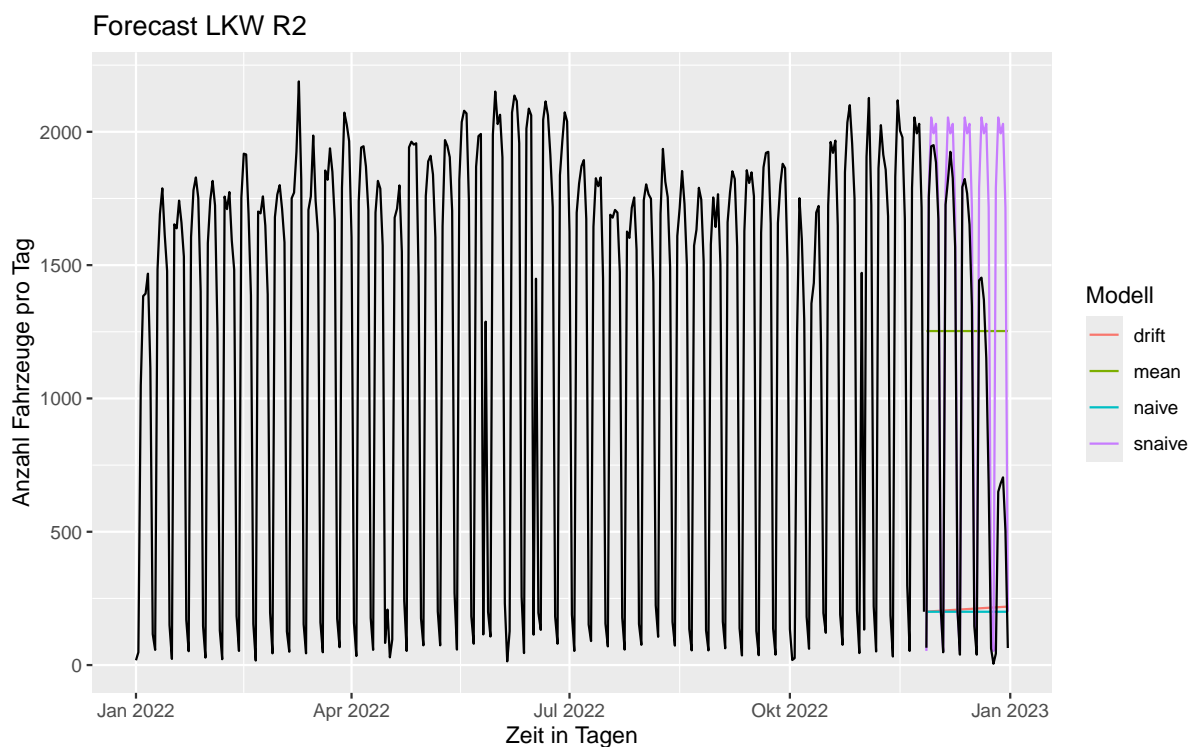
```
3 autolayer(test_pkw_R2, Pkw_Gruppe_R2, color = "black") +  
4 labs(title = "Forecast PKW R2",  
5       y = "Anzahl Fahrzeuge pro Tag",  
6       x = "Zeit in Tagen") +  
7 guides(color = guide_legend(title = "Modell"))
```



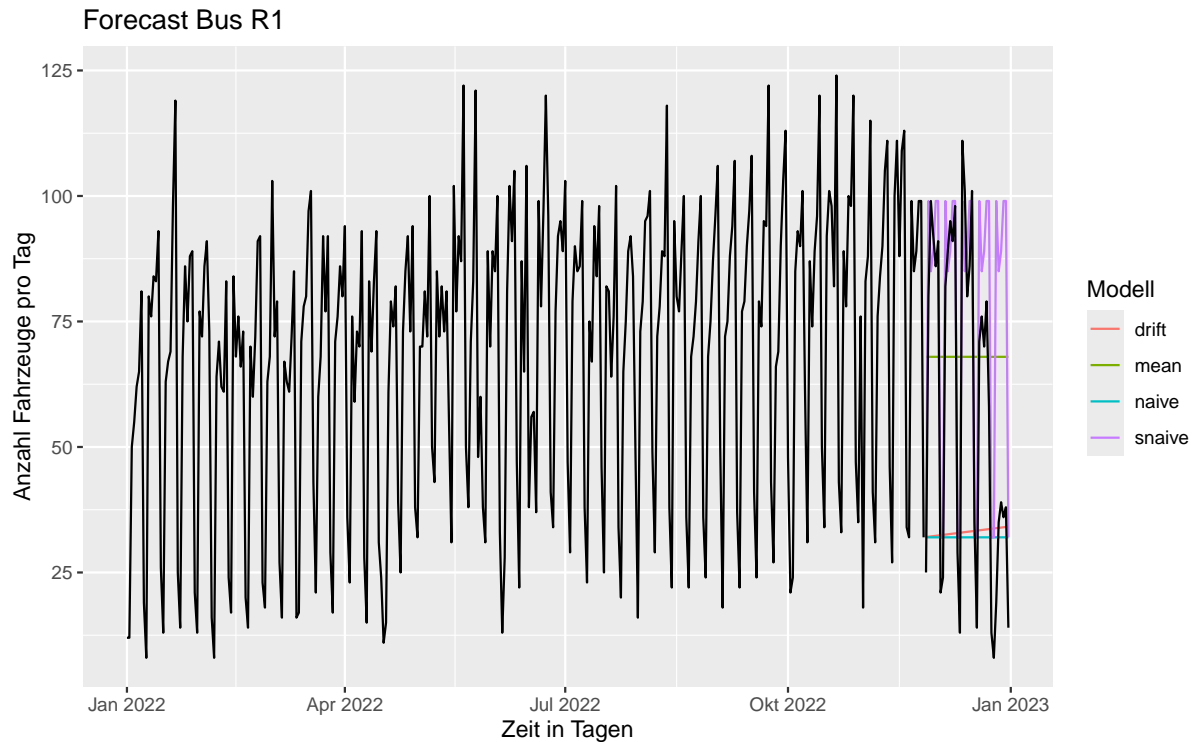
```
1 # LKW  
2 fc_lkw_R1 %>%  
3 autoplot(train_lkw_R1, level = NULL) +  
4 autolayer(test_lkw_R1, Lkw_mA_R1, color = "black") +  
5 labs(title = "Forecast LKW R1",  
6       y = "Anzahl Fahrzeuge pro Tag",  
7       x = "Zeit in Tagen") +  
8 guides(color = guide_legend(title = "Modell"))
```



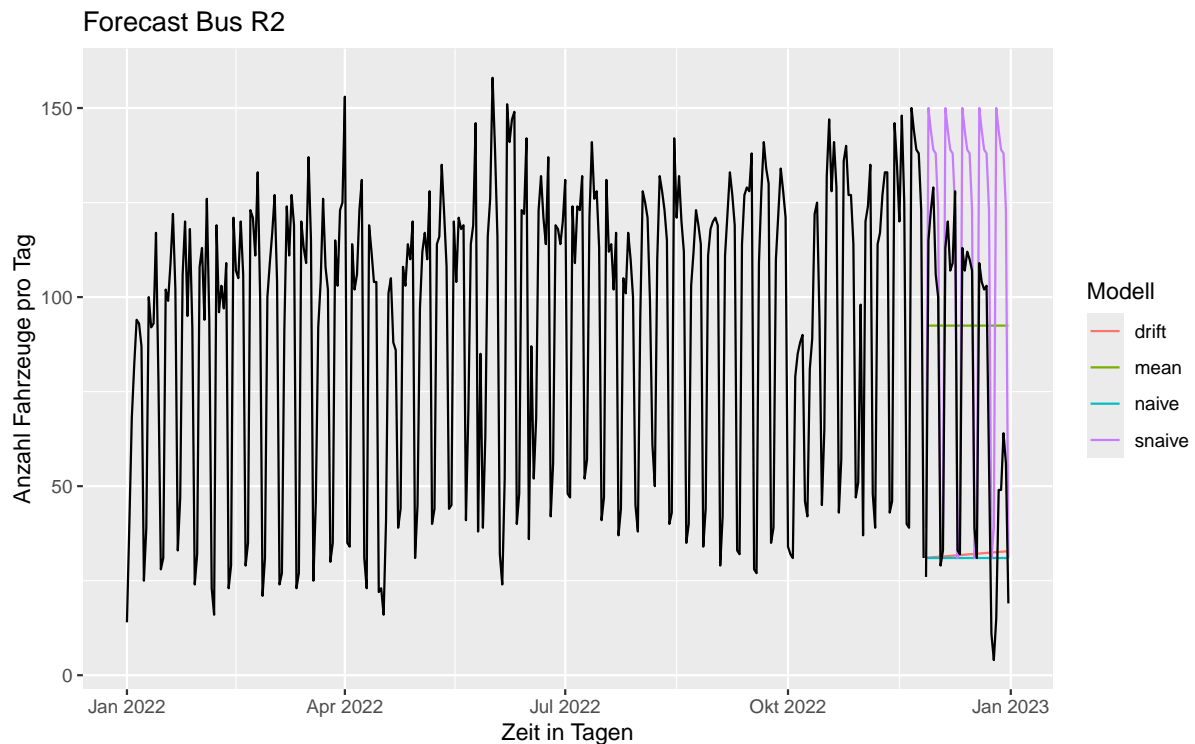
```
1 fc_lkw_R2 %>%  
2   autoplot(train_lkw_R2, level = NULL) +  
3   autolayer(test_lkw_R2, Lkw_mA_R2, color = "black") +  
4   labs(title = "Forecast LKW R2",  
5         y = "Anzahl Fahrzeuge pro Tag",  
6         x = "Zeit in Tagen") +  
7   guides(color = guide_legend(title = "Modell"))
```



```
1 # Bus
2 fc_bus_R1 %>%
3   autoplot(train_bus_R1, level = NULL) +
4   autolayer(test_bus_R1, Busse_R1, color = "black") +
5   labs(title = "Forecast Bus R1",
6         y = "Anzahl Fahrzeuge pro Tag",
7         x = "Zeit in Tagen") +
8   guides(color = guide_legend(title = "Modell"))
```



```
1 fc_bus_R2 %>%
2   autoplot(train_bus_R2, level = NULL) +
3   autolayer(test_bus_R2, Busse_R2, color = "black") +
4   labs(title = "Forecast Bus R2",
5         y = "Anzahl Fahrzeuge pro Tag",
6         x = "Zeit in Tagen") +
7   guides(color = guide_legend(title = "Modell"))
```



### 3.2.2 Metriken

```
1 #accuracy(fc_pkw_lkw_bus, test_pkw_lkw_bus)
2
3 acc_pkw_R1 <- accuracy(fc_pkw_R1, test_pkw_R1)
4 acc_pkw_R2 <- accuracy(fc_pkw_R2, test_pkw_R2)
5
6 acc_lkw_R1 <- accuracy(fc_lkw_R1, test_lkw_R1)
7 acc_lkw_R2 <- accuracy(fc_lkw_R2, test_lkw_R2)
8
9 acc_bus_R1 <- accuracy(fc_bus_R1, test_bus_R1)
10 acc_bus_R2 <- accuracy(fc_bus_R2, test_bus_R2)
11
12 list(acc_pkw_R1, acc_pkw_R2, acc_lkw_R1, acc_lkw_R2, acc_bus_R1, acc_bus_R2)
```

```
## [[1]]
## # A tibble: 4 x 10
##   .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 drift  Test    606. 1543. 1198.  4.31 15.4   NaN   NaN  0.131
## 2 mean   Test   -13.6 1413. 1025. -3.85 14.3   NaN   NaN  0.126
## 3 naive  Test    773. 1611. 1288.  6.54 16.3   NaN   NaN  0.126
## 4 snaive Test   -347. 1201. 1004. -5.04 13.8   NaN   NaN  0.192
##
## [[2]]
## # A tibble: 4 x 10
##   .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
```



```
## 1 drift Test 633. 1352. 1125. 5.15 15.8 NaN NaN 0.251
## 2 mean Test -120. 1183. 804. -4.97 12.7 NaN NaN 0.234
## 3 naive Test 736. 1388. 1177. 6.60 16.3 NaN NaN 0.234
## 4 snaive Test -541. 1170. 899. -9.56 13.6 NaN NaN 0.446
##
## [[3]]
## # A tibble: 4 x 10
##   .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
##   <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 drift Test 838. 1148. 935. -109. 221. NaN NaN 0.512
## 2 mean Test -228. 814. 717. -1010. 1036. NaN NaN 0.508
## 3 naive Test 850. 1155. 939. -95.8 209. NaN NaN 0.508
## 4 snaive Test -364. 611. 374. -171. 172. NaN NaN 0.618
##
## [[4]]
## # A tibble: 4 x 10
##   .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
##   <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 drift Test 819. 1111. 899. -131. 245. NaN NaN 0.501
## 2 mean Test -224. 781. 691. -1255. 1281. NaN NaN 0.498
## 3 naive Test 829. 1116. 903. -116. 232. NaN NaN 0.498
## 4 snaive Test -376. 616. 378. -196. 198. NaN NaN 0.652
##
## [[5]]
## # A tibble: 4 x 10
##   .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
##   <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 drift Test 26.9 42.8 35.5 3.88 68.2 NaN NaN 0.455
## 2 mean Test -7.99 34.0 30.1 -96.2 120. NaN NaN 0.446
## 3 naive Test 28.0 43.3 36.0 7.60 67.1 NaN NaN 0.446
## 4 snaive Test -16.5 27.5 19.9 -63.3 67.0 NaN NaN 0.628
##
## [[6]]
## # A tibble: 4 x 10
##   .model .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
##   <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 drift Test 44.2 60.1 49.3 13.0 77.5 NaN NaN 0.484
## 2 mean Test -16.4 43.7 37.0 -150. 168. NaN NaN 0.479
## 3 naive Test 45.1 60.7 49.8 16.2 76.3 NaN NaN 0.479
## 4 snaive Test -33.1 44.6 33.9 -95.4 97.2 NaN NaN 0.574
```

### 3.3 Auswertung

Geringster RMSE pro Klasse und Richtung: - PKW R1: Snaive - PKW R2: Snaive - LKW R1: Snaive - LKW R2: Mean (aber kaum besser als Snaive) - Bus R1: Snaive - Bus R2: Mean (aber kaum besser als Snaive) Bei den PKW ist die saisonale Methode, die mit dem geringsten RMSE in beide Richtungen. Bei den LKW und Bussen Richtung 1 zeigt Snaive ebenfalls die geringsten Werte auf. In Richtung 2 weist die Mean Methode leicht bessere Werte als die Snaive Methode auf.

### 3.4 Anmerkung

Aufgrund der ähnlichen aber entgegengesetzten Dynamiken der beiden Richtungen (z.B. freitags stärkerer Verkehr in Richtung 1 und sonntags stärkerer Verkehr in Richtung 2) und aufgrund von Zeit und Aufwand, ohne dadurch relevante Erkenntnisse gewinnen zu würden, wird im Folgenden nur noch mit den drei Fahrzeugklassen PKW, LKW mit Anhänger und Bus in Richtung 2 gearbeitet. Richtung 2 wird aufgrund der geringeren Streuung gewählt.

## 4 ETS

### 4.1 Theretische Überlegungen

- Error
  - N: kein Sinn, da Fehler vorhanden
  - A: sinnvoll da gleichmäßiger Varianz
  - M: sinnvoll, wenn Varianz mit Level steigt (eher nicht der Fall, aber testen)
- Trend
  - N: sinnvoll, da Trend kaum vorhanden, eher Saisonalität
  - A: - (wäre sinnvoll bei linearem Wachstum)
  - Ad:- (wäre sinnvoll bei linearem Wachstum mit Dämpfung)
  - M: - (wäre sinnvoll bei exp. Wachstum)
- Saisonalität
  - N: - (da Saisonalität vorhanden)
  - A: sinnvoll, da sich saisonale Effekte (Sommer/Winter) zum Niveau addieren
  - M: sinnvoll, wenn sich saisonale Effekte prozentual wirken (eher nicht der Fall, aber testen)

-> Folgende ETS() Modelle testen: - ANA -> beste Erfolgchancen - ANM - MNA - MNM

#### 4.1.1 Bedingungen erfüllt?

- keine Abrupten Änderungen
- keine extremen Ausreißer(/auffällige einzelne Werte) erkennbar
- stabile Varianz
- aber Feiertage, Ferien etc. erkennbar (seperat berücksichtigen oder erweitern -> ETSX?)

### 4.2 Praxis

#### 4.2.1 Modelle

```
1 # PKW R2
2 fit_pkw_R2 <- pkw_R2_ts %>% #fit_pkw_R2_ets besser
3   model(
4     ANA = ETS(Pkw_Gruppe_R2 ~ error("A") + trend("N") + season("A")),
5     ANM = ETS(Pkw_Gruppe_R2 ~ error("A") + trend("N") + season("M")),
6     MNA = ETS(Pkw_Gruppe_R2 ~ error("M") + trend("N") + season("A")),
7     MNM = ETS(Pkw_Gruppe_R2 ~ error("M") + trend("N") + season("M"))
8   )
```

```
9
10 # LKW R2
11 fit_lkw_R2 <- lkw_R2_ts %>%
12   model(
13     ANA = ETS(Lkw_mA_R2 ~ error("A") + trend("N") + season("A")),
14     ANM = ETS(Lkw_mA_R2 ~ error("A") + trend("N") + season("M")),
15     MNA = ETS(Lkw_mA_R2 ~ error("M") + trend("N") + season("A")),
16     MNM = ETS(Lkw_mA_R2 ~ error("M") + trend("N") + season("M"))
17   )
18
19 # Bus R2
20 fit_bus_R2 <- bus_R2_ts %>%
21   model(
22     ANA = ETS(Busse_R2 ~ error("A") + trend("N") + season("A")),
23     ANM = ETS(Busse_R2 ~ error("A") + trend("N") + season("M")),
24     MNA = ETS(Busse_R2 ~ error("M") + trend("N") + season("A")),
25     MNM = ETS(Busse_R2 ~ error("M") + trend("N") + season("M"))
26   )
```

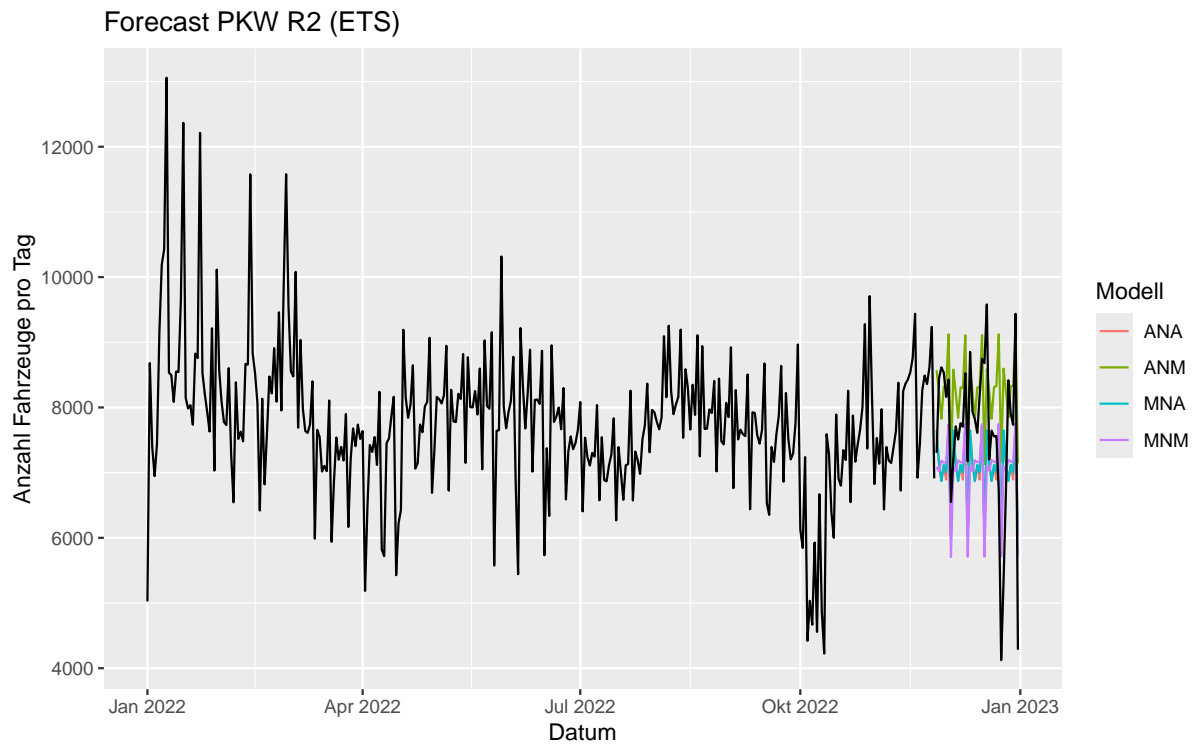
#### 4.2.2 Forecasts

```
1 fc_pkw_R2_ets <- fit_pkw_R2 %>%
2   forecast(new_data = test_pkw_R2)
3
4 fc_lkw_R2_ets <- fit_lkw_R2 %>%
5   forecast(new_data = test_lkw_R2)
6
7 fc_bus_R2_ets <- fit_bus_R2 %>%
8   forecast(new_data = test_bus_R2)
```

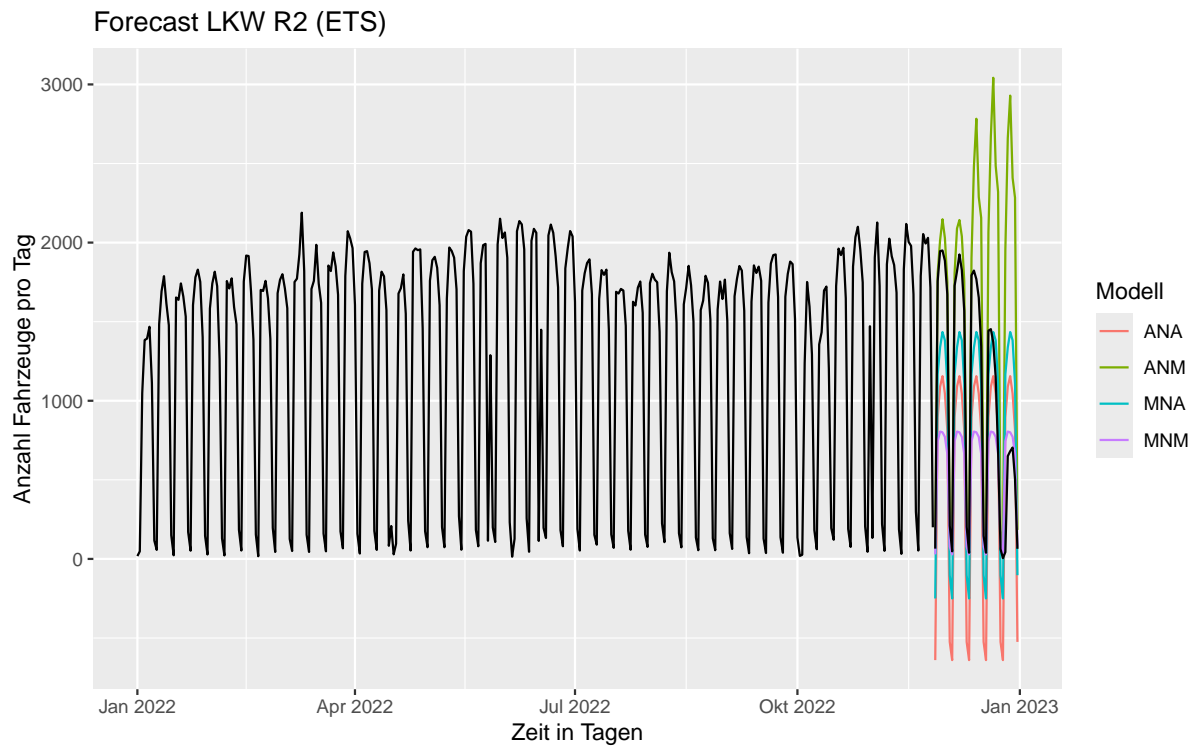
#### 4.2.3 Plots

(Für grobe Vorstellung vorerst ohne Kofidenzintervalle -> später dann mit)

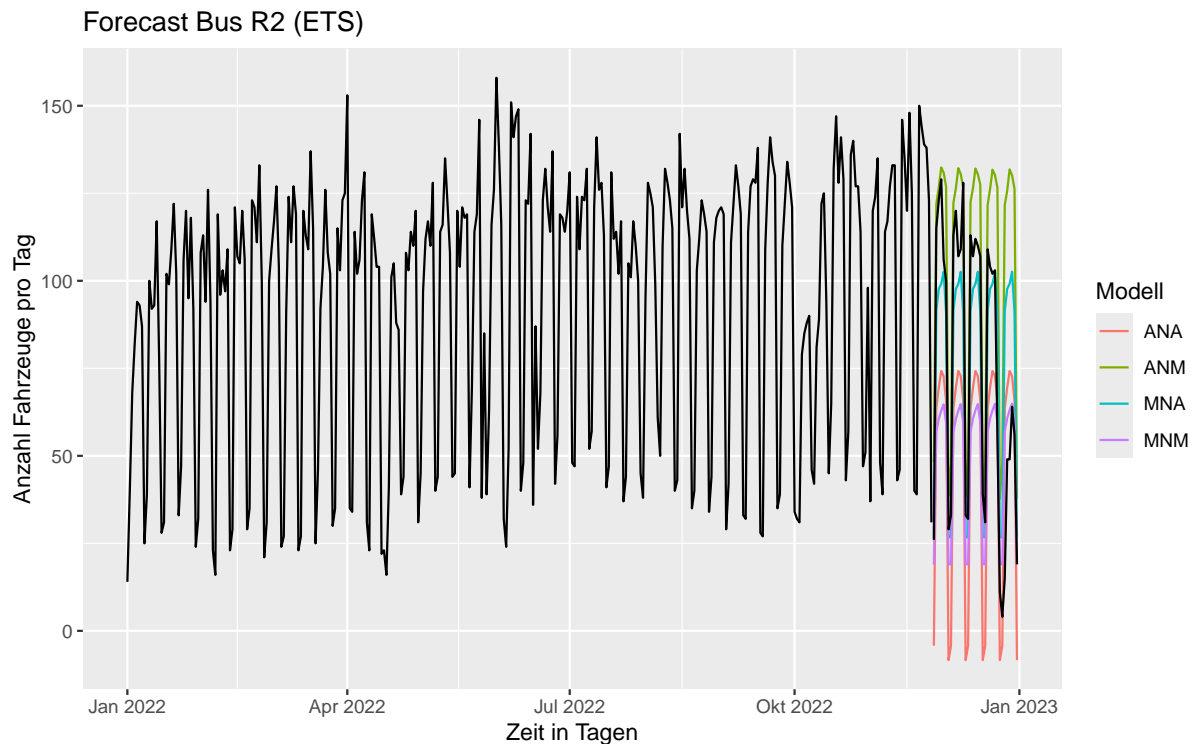
```
1 #fc_pkw_R2_ets <- fit_pkw_R2 %>%
2   #forecast(new_data = test_pkw_R2, level = c(80, 95))
3
4 # PKW R2
5 fc_pkw_R2_ets %>%
6   autoplot(train_pkw_R2, level = NULL) +
7   autolayer(test_pkw_R2, Pkw_Gruppe_R2, color = "black") +
8   labs(title = "Forecast PKW R2 (ETS)",
9         y = "Anzahl Fahrzeuge pro Tag",
10        x = "Datum") +
11   guides(color = guide_legend(title = "Modell"))
```



```
1 # LKW R2
2 fc_lkw_R2_ets %>%
3   autoplot(train_lkw_R2, level = NULL) +
4   autolayer(test_lkw_R2, Lkw_mA_R2, color = "black") +
5   labs(title = "Forecast LKW R2 (ETS)",
6         y = "Anzahl Fahrzeuge pro Tag",
7         x = "Zeit in Tagen") +
8   guides(color = guide_legend(title = "Modell"))
```



```
1 # Bus R2
2 fc_bus_R2_ets %>%
3   autoplot(train_bus_R2, level = NULL) +
4   autolayer(test_bus_R2, Busse_R2, color = "black") +
5   labs(title = "Forecast Bus R2 (ETS)",
6        y = "Anzahl Fahrzeuge pro Tag",
7        x = "Zeit in Tagen") +
8   guides(color = guide_legend(title = "Modell"))
```



## 4.3 Bewertung der Modelgüte

### 4.3.1 AIC, AICc und BIC

```
1 report(fit_pkw_R2)
```

```
## # A tibble: 4 x 9
##   .model      sigma2 log_lik   AIC   AICc   BIC     MSE    AMSE    MAE
##   <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 ANA      791365.    -3551. 7122. 7122. 7161. 771852. 900430. 598.
## 2 ANM      762782.    -3544. 7108. 7109. 7147. 743974. 880620. 607.
## 3 MNA         0.0133 -3551. 7123. 7123. 7162. 775125. 888162. 0.0773
## 4 MNM         0.0130 -3548. 7116. 7117. 7155. 756269. 856650. 0.0792
```

```
1 report(fit_lkw_R2)
```

```
## # A tibble: 4 x 9
##   .model      sigma2 log_lik   AIC   AICc   BIC     MSE    AMSE    MAE
##   <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1 ANA      82357.    -3138. 6296. 6296. 6335. 80326. 85839. 147.
## 2 ANM      72679.    -3115. 6250. 6251. 6289. 70887. 81684. 130.
## 3 MNA         0.334 -3287. 6593. 6594. 6632. 90779. 93098. 0.268
## 4 MNM         0.0672 -2991. 6003. 6003. 6042. 74040. 82121. 0.159
```

```
1 report(fit_bus_R2)
```

```
## # A tibble: 4 x 9
##   .model      sigma2 log_lik   AIC   AICc   BIC     MSE    AMSE    MAE
```

```
##   <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ANA      310.     -2119. 4258. 4259. 4297.  302.  333. 11.7
## 2 ANM      305.     -2116. 4253. 4253. 4292.  298.  327. 11.5
## 3 MNA       0.0529 -2140. 4301. 4301. 4340.  397.  399.  0.161
## 4 MNM       0.0466 -2119. 4258. 4258. 4297.  307.  334.  0.146
```

#### 4.3.2 MAE und RMSE

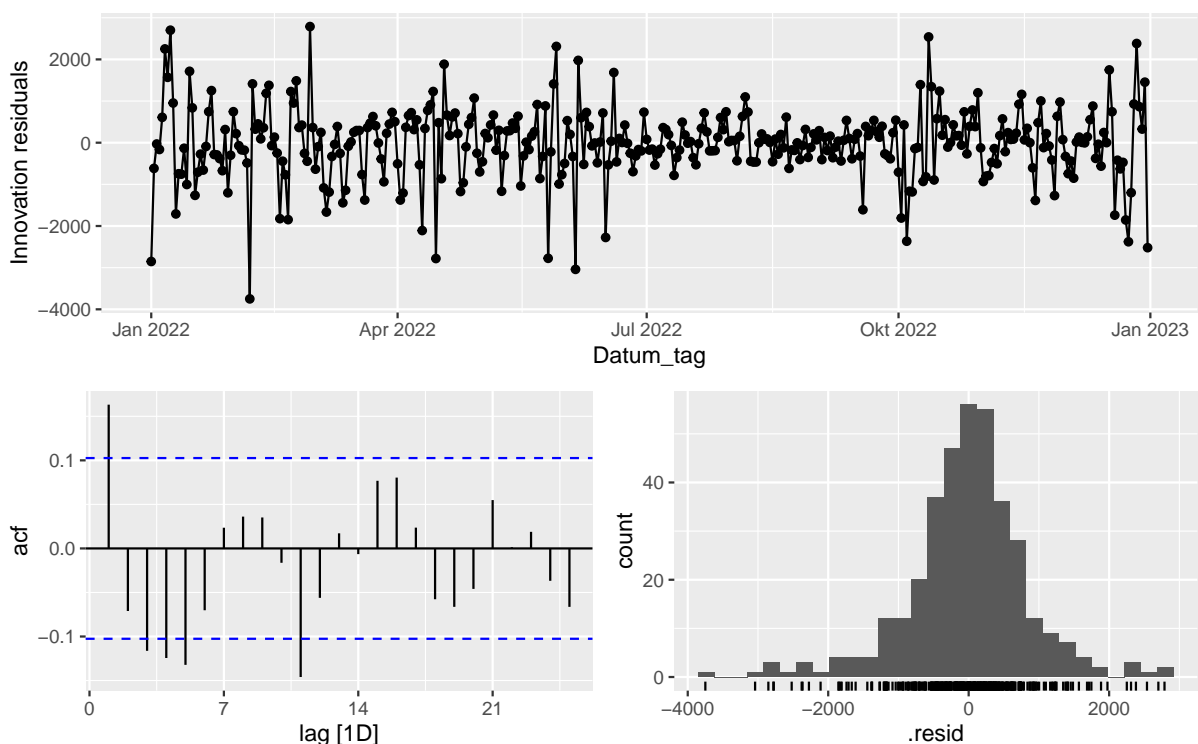
```
1  # In-sample accuracy (Trainingsdaten)
2  acc_pkw_R2_ets_train <- accuracy(fit_pkw_R2)
3  acc_lkw_R2_ets_train <- accuracy(fit_lkw_R2)
4  acc_bus_R2_ets_train <- accuracy(fit_bus_R2)
5
6  # Out-of sample accuracy (Forecast, Testdaten)
7  acc_pkw_R2_ets_test  <- accuracy(fc_pkw_R2_ets, test_pkw_R2)
8  acc_lkw_R2_ets_test  <- accuracy(fc_lkw_R2_ets, test_lkw_R2)
9  acc_bus_R2_ets_test  <- accuracy(fc_bus_R2_ets, test_bus_R2)
10
11 list(acc_pkw_R2_ets_train, acc_lkw_R2_ets_train, acc_bus_R2_ets_train, acc_pkw_R2_ets_test, acc_lkw_R2_ets_test, acc_bus_R2_ets_test)

## [[1]]
## # A tibble: 4 x 10
##   .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ANA    Training -13.1  879.  598. -1.29  8.05  0.822  0.764  0.181
## 2 ANM    Training -21.3  863.  607. -1.35  8.29  0.834  0.750  0.163
## 3 MNA    Training -15.4  880.  599. -1.36  8.08  0.823  0.765  0.208
## 4 MNM    Training -25.2  870.  611. -1.48  8.37  0.839  0.756  0.227
##
## [[2]]
## # A tibble: 4 x 10
##   .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ANA    Training  -8.48  283.  147. -40.2 109.  0.867  0.724  0.123
## 2 ANM    Training  -4.67  266.  130. -59.4  73.5  0.767  0.680  0.0949
## 3 MNA    Training -14.2  301.  152. -53.8  82.3  0.901  0.770  0.286
## 4 MNM    Training  19.4  272.  138. -53.5  65.7  0.817  0.695  0.172
##
## [[3]]
## # A tibble: 4 x 10
##   .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ANA    Training -0.345  17.4  11.7 -8.07  21.6  0.796  0.739  0.103
## 2 ANM    Training -0.243  17.3  11.5 -9.75  21.2  0.788  0.734  0.194
## 3 MNA    Training -0.00748 19.9  13.3 -10.8  24.0  0.907  0.847  0.419
## 4 MNM    Training -0.675  17.5  11.6 -9.69  21.1  0.795  0.744  0.221
##
## [[4]]
## # A tibble: 4 x 10
##   .model .type      ME  RMSE  MAE  MPE  MAPE  MASE  RMSSE  ACF1
```

```
##   <chr>   <chr> <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ANA     Test   600. 1208. 1053.    5.47  14.5   NaN   NaN  0.429
## 2 ANM     Test  -573. 1195.  856.   -10.3  13.5   NaN   NaN  0.430
## 3 MNA     Test   565. 1193. 1036.    4.99  14.4   NaN   NaN  0.426
## 4 MNM     Test   635. 1176.  985.    6.22  13.3   NaN   NaN  0.465
##
## [[5]]
## # A tibble: 4 x 10
##   .model .type    ME RMSE  MAE    MPE  MAPE  MASE RMSSE  ACF1
##   <chr>   <chr> <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ANA     Test   481.  661.  627.   583.  712.   NaN   NaN  0.710
## 2 ANM     Test  -610.  939.  612.  -252.  253.   NaN   NaN  0.667
## 3 MNA     Test   161.  472.  422.   171.  350.   NaN   NaN  0.737
## 4 MNM     Test   473.  705.  545.  -25.4  102.   NaN   NaN  0.648
##
## [[6]]
## # A tibble: 4 x 10
##   .model .type    ME RMSE  MAE    MPE  MAPE  MASE RMSSE  ACF1
##   <chr>   <chr> <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 ANA     Test    28.5  38.0  35.5   45.8  72.4   NaN   NaN  0.737
## 2 ANM     Test   -27.3  37.6  27.4  -97.2  97.4   NaN   NaN  0.673
## 3 MNA     Test   -0.387  24.9  18.5  -37.2  57.2   NaN   NaN  0.749
## 4 MNM     Test    27.1  39.1  32.8    9.56  55.0   NaN   NaN  0.656
```

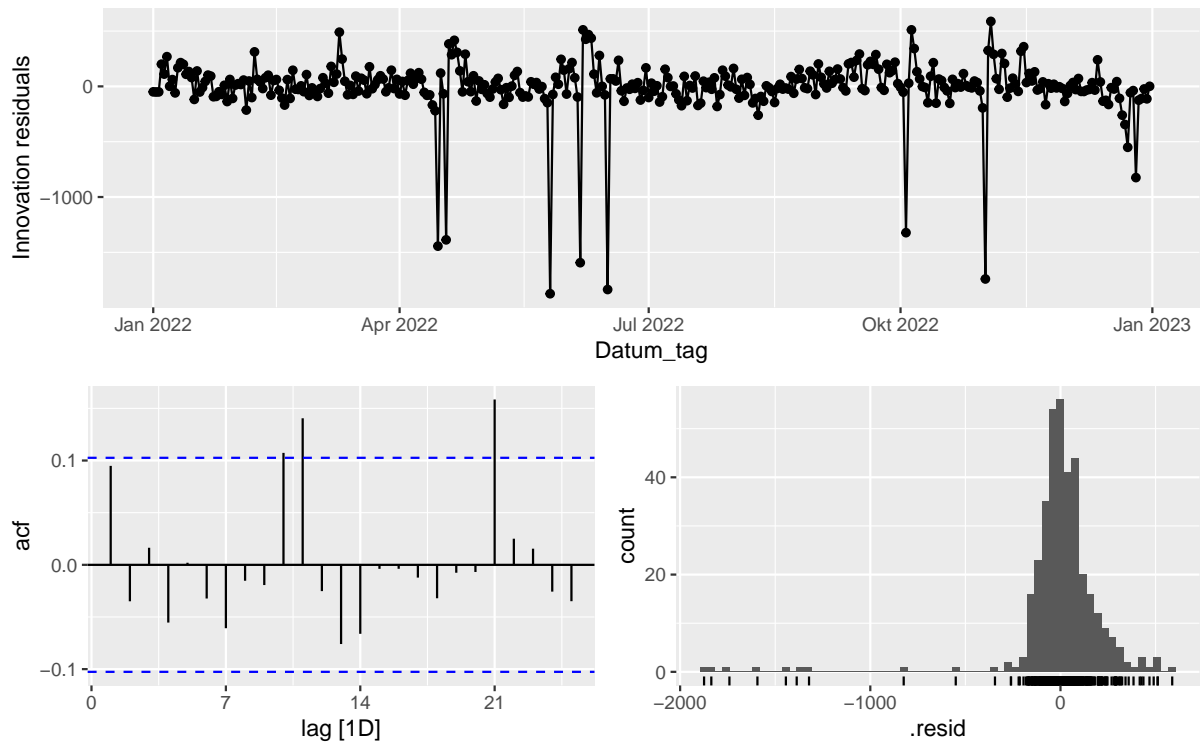
### 4.3.3 Residuenanalyse (Hist, ACF, Ljung Box)

```
1 fit_pkw_R2 %>% select(ANM) %>% gg_tsresiduals()
```

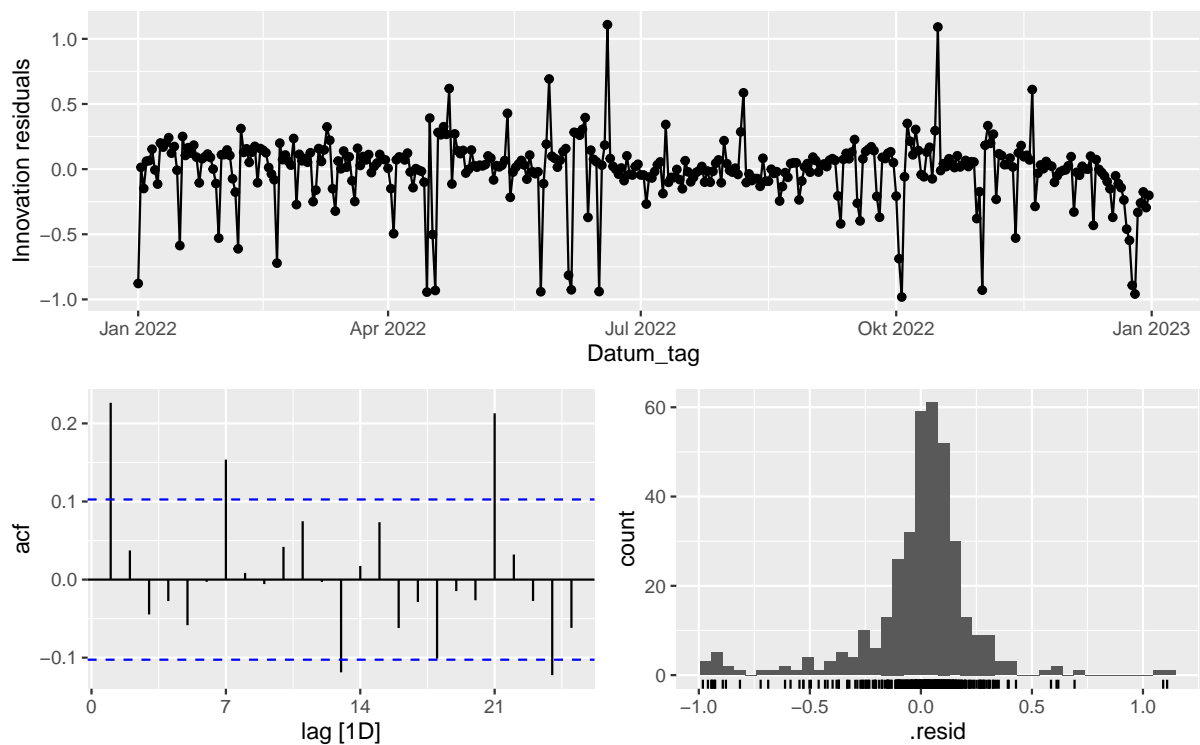




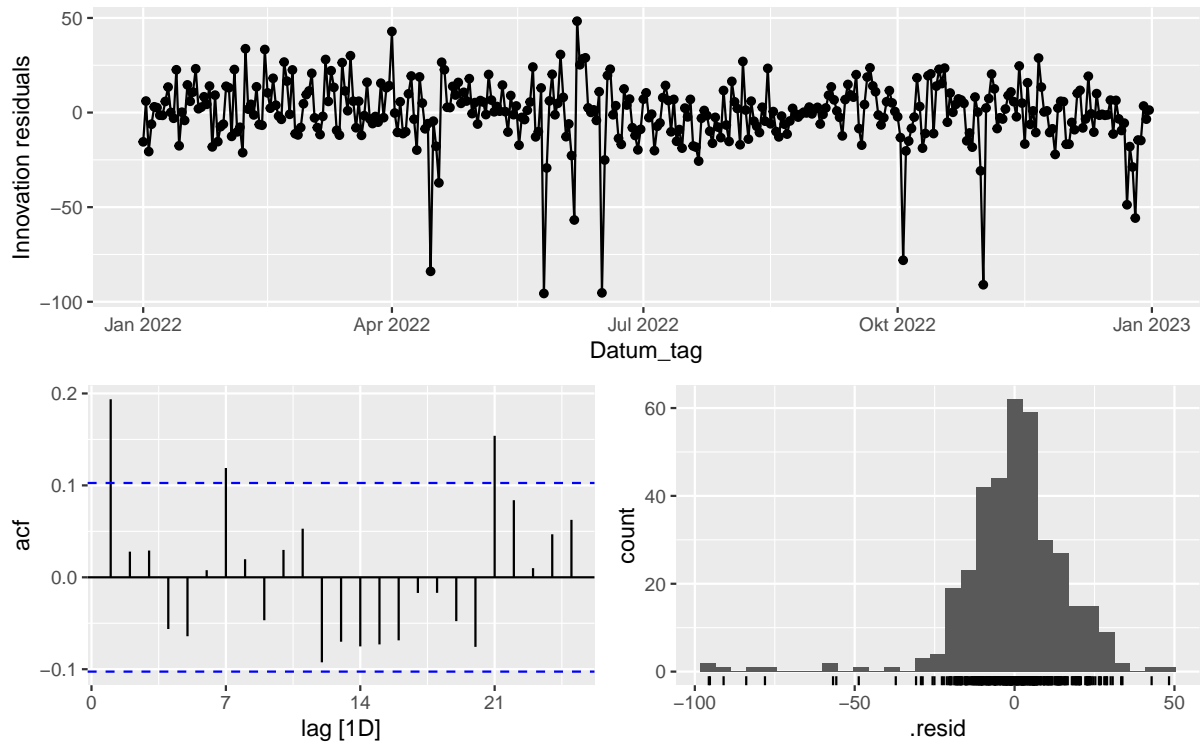
```
1 fit_lkw_R2 %>% select(ANM) %>% gg_tsresiduals()
```



```
1 fit_lkw_R2 %>% select(MNM) %>% gg_tsresiduals() #besser
```



```
1 fit_bus_R2 %>% select(ANM) %>% gg_tsresiduals()
```



```
1 # PKW ETS(ANM)
2 #Box.test(res_pkw$.resid, type = "Ljung-Box", lag = 20)
3 # LKW ETS(ANM)
4 #Box.test(res_lkw_mnm$.resid, type = "Ljung-Box", lag = 20)
5 # LKW ETS(MNM)
6 #Box.test(res_lkw_anm$.resid, type = "Ljung-Box", lag = 20)
7 # Bus ETS(ANM)
8 #Box.test(res_bus$.resid, type = "Ljung-Box", lag = 20)
```

Aufgrund von Fehlern beim Knitten ist der Ljung Box Test hier nur auskommentiert. Der Code läuft aber und auf die Ergebnisse wird im Folgenden eingegangen.

## 4.4 Erkenntnisse

### 4.4.1 Geringste AIC(c) und BIC Werte in jeweiliger Klasse:

- PKW: ANM (-> gut weil Varianz stabil)
- LKW: MNM (-> stärkerer Schwankungen proportional zum Level)
- Bus: ANM

### 4.4.2 MAE und RMSE

Wo RMSE (und MAE) am Kleinsten (In Sample): - PKW: ANM (MAE bei ANA am Kleinsten) - LKW: ANM - Bus: ANM

Out Sample: - PKW: ANM - LKW: ANM - Bus: ANM (MAE bei MNA am Kleinsten)

#### 4.4.3 Residuenanalyse

PKW ETS(ANM) - Zeitreihe der Residuen: schwanken um Null, aber größere Ausschläge sichtbar (besonder durch Feiertage: Ostern, Tag der Deutschen Einheit, Weihnachte etc. sind klar erkennbar) - ACF Plot: Werte schlagen bei den PKW zu Beginn über die Intervallgrenzen nach unten aus und einmal bei Lag 11 nach oben; insgesamt aber kein starkes Muster zu erkennen (keine systematische Autokorrelation) - Histogramm: normalverteilt, links etwas länger als rechts (Modell unterschätzt PKW Anzahl an manchen Tagen)

Das Modell erfasst einen Großteil der Varianz (aber keine Feiertage) und es erklärt die Abhängigkeit der Serie recht gut (fast alle Werte liegen innerhalb der Konfidenzintervalle). Das Histogramm zeigt mit seiner rechtsschiefen Verteilung, dass das Modell die Anzahl der PKW an einigen Tagen unterschätzt, was sich mit der Zeitreihe der Residuen deckt (Feiertage -> mehr Verkehr).

LKW ETS(ANM) - Zeitreihe der Residuen: schwanken insgesamt um Null, aber auch hier Werte an Feiertagen die das Modell unterschätzt (Ostern, Christi Himmelfahrt, Pfingsten und Frohenleichnam -> auch Reformationstag fällt auf, obwohl in NRW kein gesetzlicher Feiertag); wirken stabil (Stabiler als PKW) - ACF Plot: Werte schlagen leicht bei 10,11 und 21 aus, ansonsten keine Auffälligkeiten (Wochensaison von Modell noch nicht erkannt) - Histogramm: weist eine grobe Normalverteilung auf, stark linksschief (mehrere kleine Werte bis weit auf der linken Seite) durch einige sehr starke negative Werte

Auch hier erfasst das Modell einen großen Teil der Varianz, die Residuen zeigen deutlich weniger Streuung als bei den PKW, aber auch hier fallen die Feiertage auf. Der ACF plot ist weitgehend unauffällig, es scheint aber so, dass das Modell die Wochensaison noch nicht ganz erfasst hat. Die Linksschiefe im Histogramm zeigt, dass das Modell die Anzahl der LKW an manchen Tagen überschätzt, was Sinn ergibt, wenn es die Wochensaison noch nicht erfasst hat, das LKW eher werktags unterwegs sind. Dies deckt sich außerdem mit den Beobachtungen in der Zeitreihe der Residuen.

Bus ETS(ANM) - Zeitreihe der Residuen: einige Werte die etwas weiter nach unten ausschlagen - ACF Plot: Lags bei 1,7 und 21 - Histogramm: normalverteilt, aber links länger als rechts

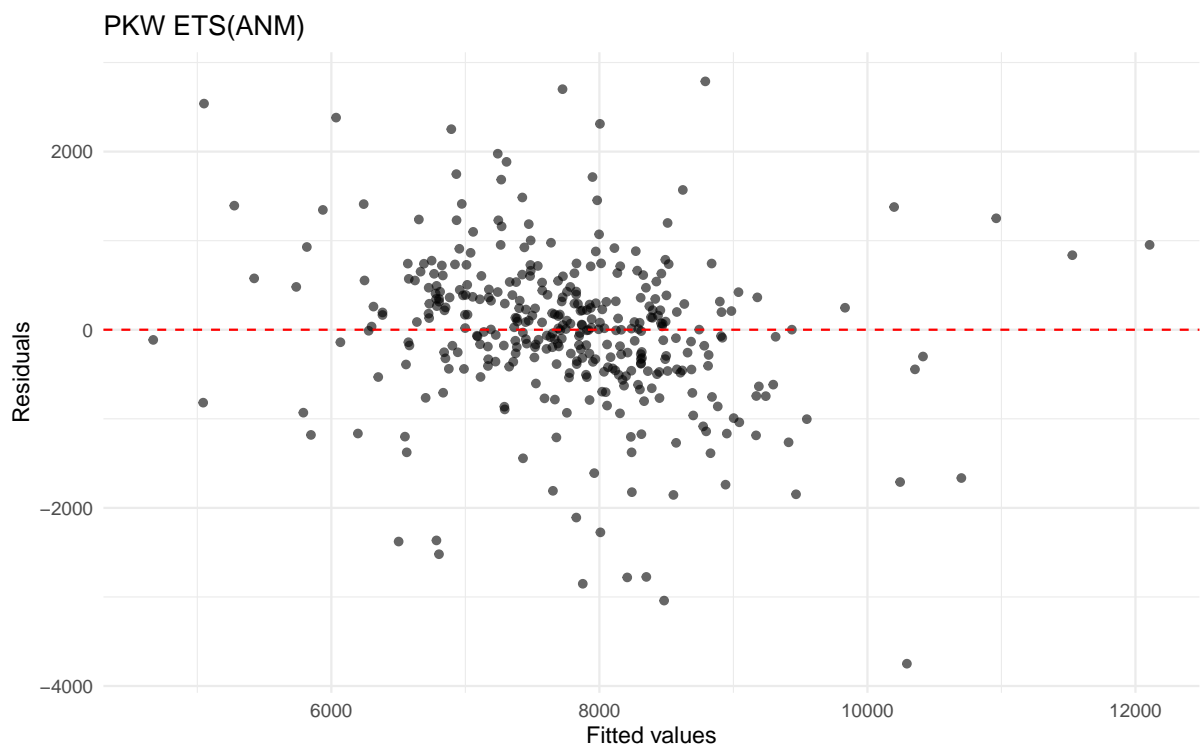
Auch hier wird die Varianz zum Großteil erfasst und die Residuen schwanken um Null. Die Feiertage machen sich ebenfalls durch auffallende negative Werte in der Zeitreihe der Residuen bemerkbar. Die Ausschläge im ACF Plot weisen auf eine schwache Autokorrelation hin, das Wochenmuster wurde noch nicht vollständig erkannt, was auch die Linksschiefe im Histogramm erklärt. Das Aufkommen an Bussen wird an manchen Tagen überschätzt, was ebenfalls Sinn ergibt, das die Buspläne sich werktags, samstags, sonntags/an Feiertagen unterscheiden.

#### 4.4.4 Ljung Box Test

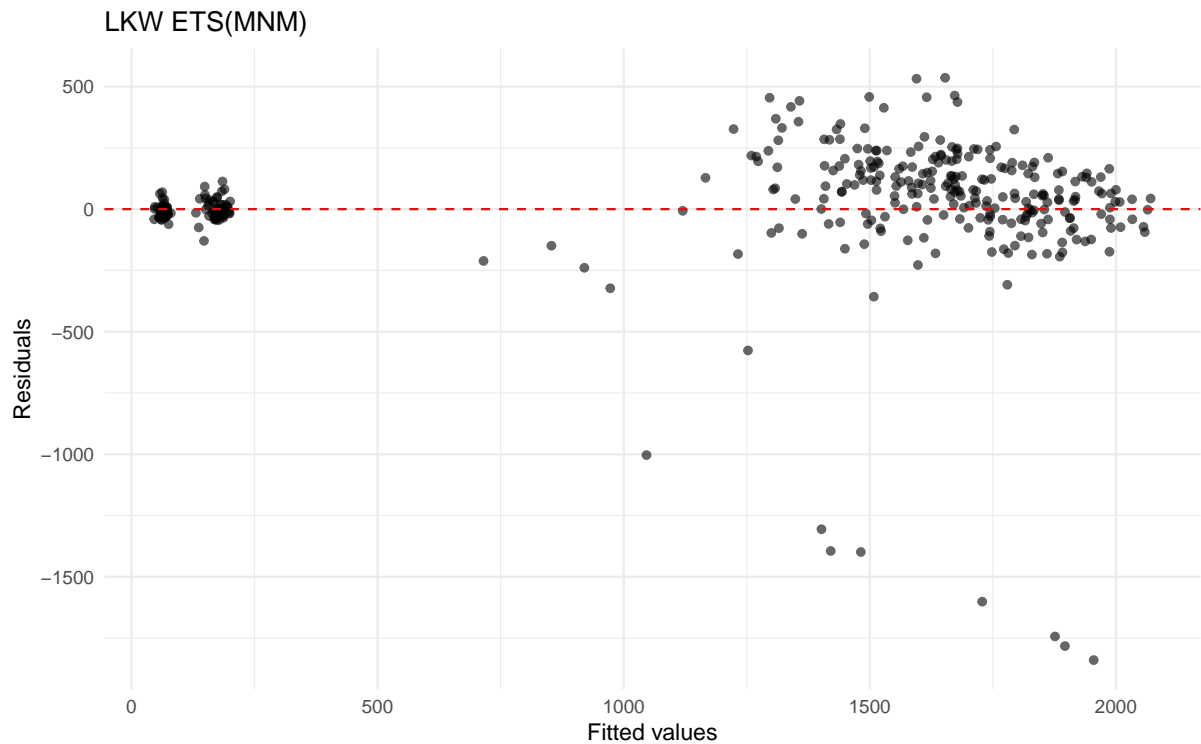
- PKW ETS(ANM): da der p value unter 0.05 liegt, wird die Nullhypothese abgelehnt. Die Residuen sind signifikant autokorreliert (das Modell erfasst noch nicht alle Strukturen).
- LKW ETS(ANM): der p value liegt über 0.05. Die Nullhypothese wird angenommen. Es liegt keine signifikante Autokorrelation vor, das Modell erfasst die Strukturen bereits gut.
- LKW ETS(MNM): Die Residuen sind autokorreliert. Das Modell erfasst (wahrscheinlich die Wochensaison) nicht vollständig.
- Bus ETS(ANM): Die Residuen sind autokorreliert. Es ist eine schwache Autokorrelation vorhanden.

#### 4.4.5 Residuen vs. fitted values

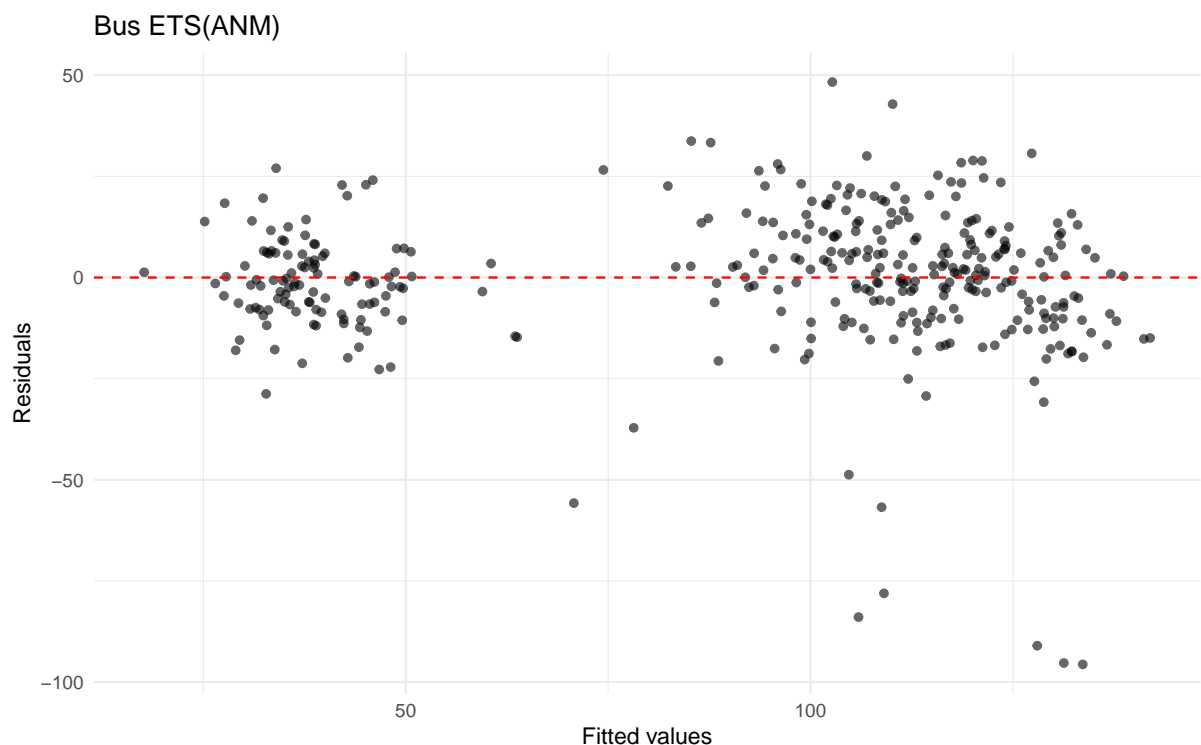
```
1 plot_resid_vs_fitted <- function(fit_obj, title = "Residuals vs Fitted") {
2   resid_df <- fit_obj %>%
3     augment() %>%
4     as.data.frame() %>%
5     select(.fitted, .resid)
6
7   ggplot(resid_df, aes(x = .fitted, y = .resid)) +
8     geom_point(alpha = 0.6) +
9     geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
10    labs(title = title,
11         x = "Fitted values",
12         y = "Residuals") +
13    theme_minimal()
14 }
15
16 plot_resid_vs_fitted(fit_pkw_R2 %>% select(ANM), "PKW ETS(ANM)")
```



```
1 plot_resid_vs_fitted(fit_lkw_R2 %>% select(MNM), "LKW ETS(MNM)")
```



```
1 plot_resid_vs_fitted(fit_bus_R2 %>% select(ANM), "Bus ETS(ANM)")
```



Es wird deutlich, dass einige Effekte (wahrscheinlich Feiertage, Wochenstruktur, Ferien) vom Modell nicht erkannt werden. Gerade bei LKW und Bussen gibt es auffällige negative Residuen, die zeigen, dass das Modell das Aufkommen an manchen Tagen überschätzt.

Bei LKW und Bussen sind einige stark negative Residuen zu erkennen, die auf eine leichte Heteroskedastizität hinweisen. Da es sich dabei jedoch nur um wenige Ausreißer handelt und

die meisten Residuen um Null streuen, wird keine Transformation der Zeitreihe vorgenommen, da der Einfluss auf das Modellverhalten minimal ist.

#### 4.4.6 $\alpha$ und $\gamma$ Werte

```
1 # PKW R2 - ETS(ANM)
2 fit_pkw_R2 %>% select(ANM) %>% tidy()
```

```
## # A tibble: 10 x 3
##   .model term      estimate
##   <chr>   <chr>      <dbl>
## 1 ANM     alpha      0.337
## 2 ANM     gamma      0.137
## 3 ANM     l[0]    8598.
## 4 ANM     s[0]       1.03
## 5 ANM     s[-1]      0.915
## 6 ANM     s[-2]      0.934
## 7 ANM     s[-3]      0.965
## 8 ANM     s[-4]      1.01
## 9 ANM     s[-5]      1.23
## 10 ANM    s[-6]      0.916
```

```
1 #report(fit_pkw_R2 %>% select(ANM))
2
3 # LKW R2 - ETS(MNM)
4 fit_lkw_R2 %>% select(MNM) %>% tidy()
```

```
## # A tibble: 10 x 3
##   .model term      estimate
##   <chr>   <chr>      <dbl>
## 1 MNM     alpha      0.147
## 2 MNM     gamma    0.000101
## 3 MNM     l[0]    1054.
## 4 MNM     s[0]       1.19
## 5 MNM     s[-1]      1.39
## 6 MNM     s[-2]      1.44
## 7 MNM     s[-3]      1.45
## 8 MNM     s[-4]      1.34
## 9 MNM     s[-5]      0.0515
## 10 MNM    s[-6]      0.140
```

```
1 #report(fit_lkw_R2 %>% select(MNM))
2
3 # Bus R2 - ETS(ANM)
4 fit_bus_R2 %>% select(ANM) %>% tidy()
```

```
## # A tibble: 10 x 3
##   .model term      estimate
##   <chr>   <chr>      <dbl>
## 1 ANM     alpha      0.169
```

```
## 2 ANM      gamma 0.000100
## 3 ANM      l[0] 80.0
## 4 ANM      s[0] 1.23
## 5 ANM      s[-1] 1.26
## 6 ANM      s[-2] 1.28
## 7 ANM      s[-3] 1.22
## 8 ANM      s[-4] 1.18
## 9 ANM      s[-5] 0.465
## 10 ANM     s[-6] 0.369
```

```
1 #report(fit_lkw_R2 %>% select(ANM))
```

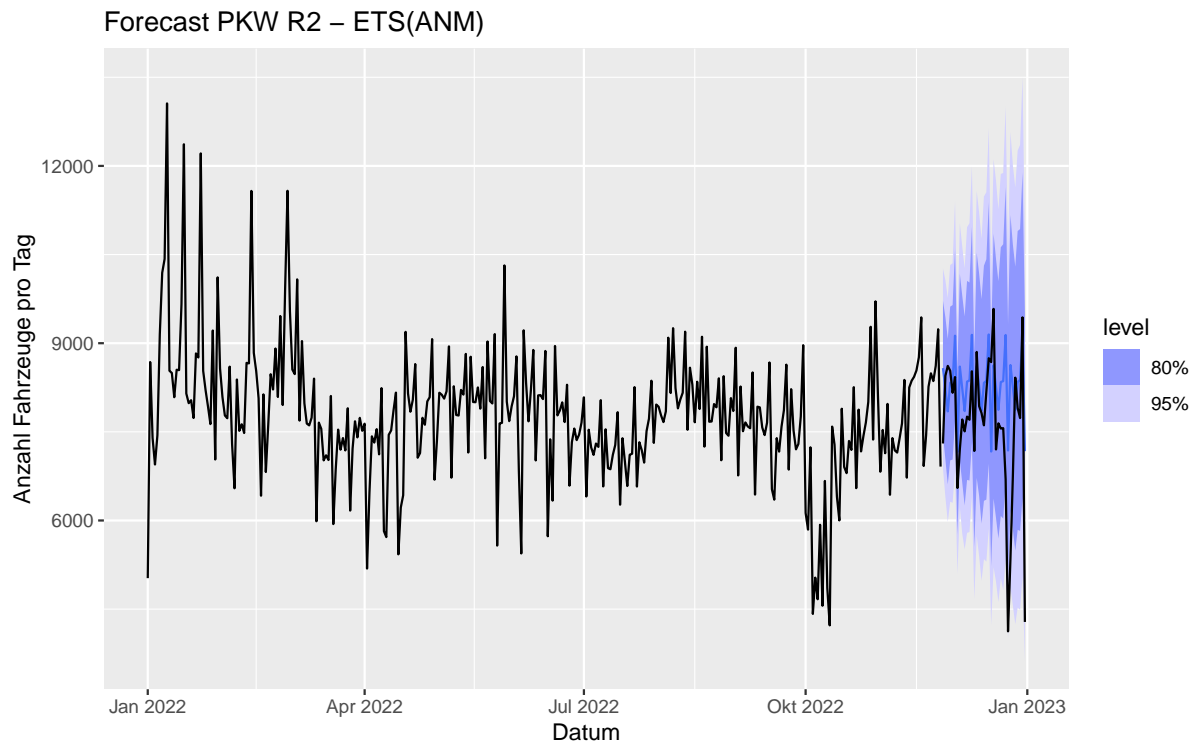
Interpretation der Glättungsparameter  $\alpha$  für die drei Modell der jeweiligen Fahrzeugklasse: - PKW ( $\alpha = 0,337$ ): Das Modell reagiert relativ stark auf neue Datenpunkte und passt das Level der Zeitreihe relativ schnell an, was bei kurzfristigen Schwankungen, wie Feiertagen Sinn ergibt. - LKW ( $\alpha = 0,147$ ): Das Modell reagiert nur leicht auf neue Beobachtungen, was dadurch zu erklären ist, dass die LKW-Zahlen stabiler sind als z.B. bei den PKW und weniger kaum kurzfristige Schwankungen haben. (-> Regelmäßigkeit durch Lieferketten) - Bus ( $\alpha = 0,169$ ): Auch hier ist der Wert relativ gering und das Modell reagiert nur leicht auf neue Beobachtungen, was ebenfalls Sinn ergibt, da z.B. durch feste Fahrpläne im ÖPNV weniger Schwankungen entstehen.

Die geschätzten Alpha-Werte zeigen, dass das PKW-Modell stärker auf kurzfristige Schwankungen reagiert, während die LKW- und Bus-Modelle eher stabile Level annehmen und weniger empfindlich auf einzelne Ausreißer reagieren.

## 4.5 Beste Modelle

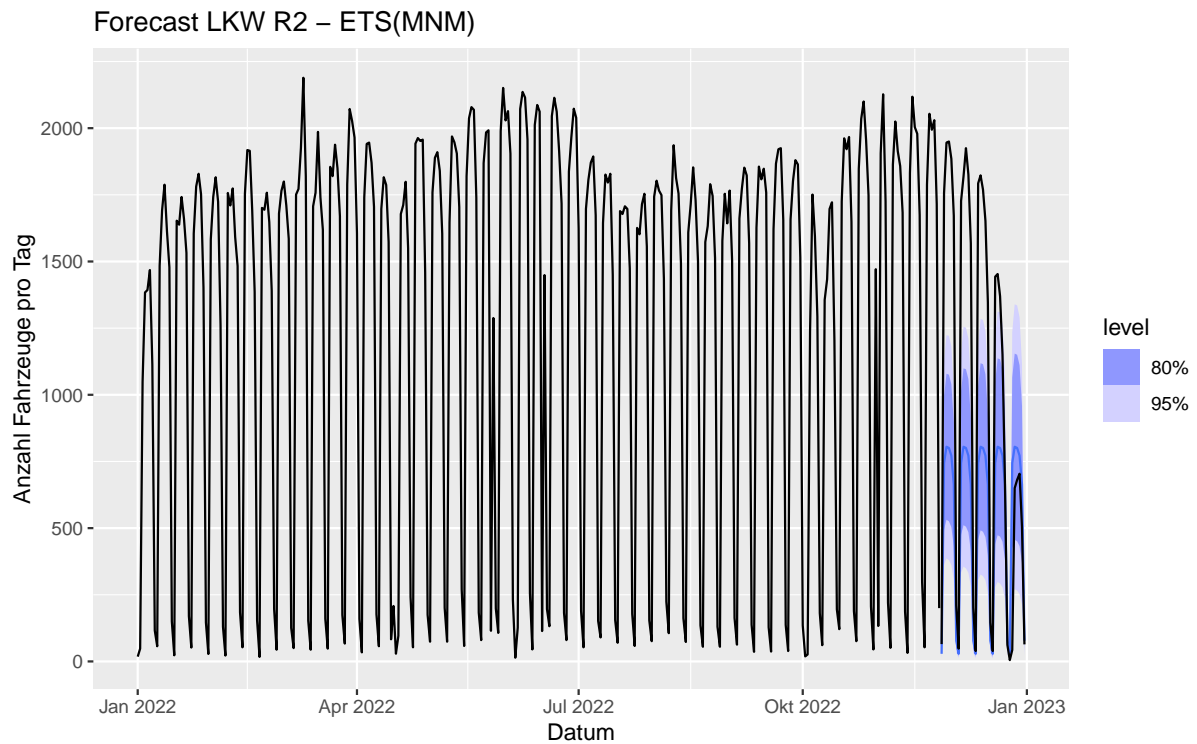
Insgesamt ist zu erkennen, dass das ETS(ANM) Modell bei PKW, LKW und Bussen die beste Anpassung aufweist, sowie die beste Prognosegüte liefern. Was über alle Modelle hinweg auffällt ist die Überschätzung des Verkehrsaufkommens an einzelnen Tagen, die meist auf Feiertage zurückzuführen sind. Im Folgenden einmal die besten Modelle für die drei Klassen:

```
1 # PKW R2 - ETS(ANM)
2 fc_pkw_R2_ets_best <- fit_pkw_R2 %>% select(ANM) %>% forecast(new_data = test_pkw_R2)
3 fc_pkw_R2_ets_best %>%
4   autoplot(train_pkw_R2, level = c(80,95)) +
5   autolayer(test_pkw_R2, Pkw_Gruppe_R2, color = "black") +
6   labs(title = "Forecast PKW R2 - ETS(ANM)",
7        y = "Anzahl Fahrzeuge pro Tag",
8        x = "Datum") +
9   guides(color = guide_legend(title = "Modell"))
```

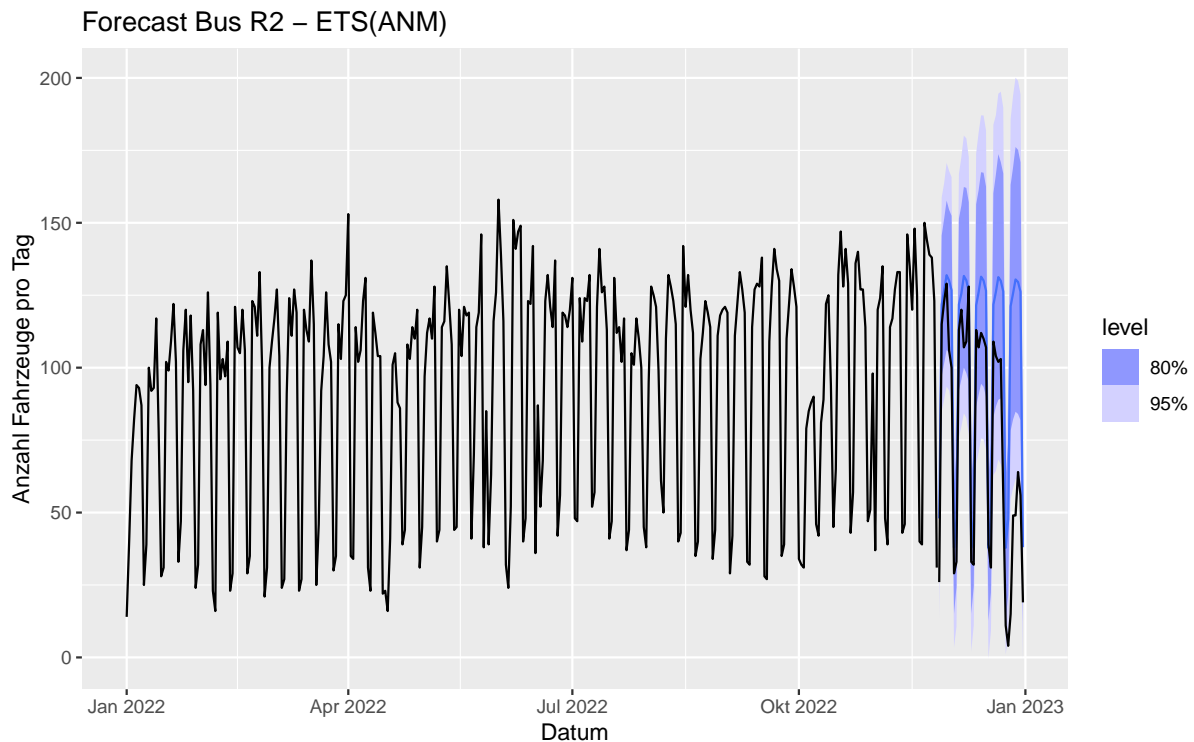


```
1 # LKW R2 - ETS(MNM)
2 fc_lkw_R2_ets_best <- fit_lkw_R2 %>% select(MNM) %>% forecast(new_data = test_lkw_R2)
3 fc_lkw_R2_ets_best %>%
4   autoplot(train_lkw_R2, level = c(80, 95)) +
5   autolayer(test_lkw_R2, Lkw_mA_R2, color = "black") +
6   labs(title = "Forecast LKW R2 - ETS(MNM)",
7         y = "Anzahl Fahrzeuge pro Tag",
8         x = "Datum") +
9   guides(color = guide_legend(title = "Modell"))
```





```
1 # Bus R2 - ETS(ANM)
2 fc_bus_R2_ets_best <- fit_bus_R2 %>% select(ANM) %>% forecast(new_data = test_bus_R2)
3 fc_bus_R2_ets_best %>%
4   autoplot(train_bus_R2, level = c(80, 95)) +
5   autolayer(test_bus_R2, Busse_R2, color = "black") +
6   labs(title = "Forecast Bus R2 - ETS(ANM)",
7        y = "Anzahl Fahrzeuge pro Tag",
8        x = "Datum") +
9   guides(color = guide_legend(title = "Modell"))
```



#### 4.5.1 Vergleich von ETS Modellen und Naiven Methoden

```
1 # CRPS für Benchmarks
2 rmse_pkw_R2_bench <- acc_pkw_R2 %>%
3   filter(.model == "snaive", .type == "Test") %>%
4
5
6   pull(RMSE)
7
8 rmse_lkw_R2_bench <- acc_lkw_R2 %>%
9   filter(.model == "mean", .type == "Test") %>%
10  pull(RMSE)
11
12 rmse_bus_R2_bench <- acc_bus_R2 %>%
13   filter(.model == "mean", .type == "Test") %>%
14   pull(RMSE)
15
16
17 # CRPS für ETS Model
18 rmse_pkw_R2_model <- acc_pkw_R2_ets_test %>% # besser ets statt model bei benennung
19   filter(.model == "ANM", .type == "Test") %>%
20   pull(RMSE)
21
22 rmse_lkw_R2_model <- acc_lkw_R2_ets_test %>%
23   filter(.model == "MNM", .type == "Test") %>%
24   pull(RMSE)
25
26 rmse_bus_R2_model <- acc_bus_R2_ets_test %>%
```

```
27 filter(.model == "ANM", .type == "Test") %>%
28 pull(RMSE)
29
30 # Skill Score
31 skill_rmse_pkw_R2 <- (rmse_pkw_R2_bench - rmse_pkw_R2_model) / rmse_pkw_R2_bench
32 skill_rmse_pkw_R2

## [1] -0.0215373

1 skill_rmse_lkw_R2 <- (rmse_lkw_R2_bench - rmse_lkw_R2_model) / rmse_lkw_R2_bench
2 skill_rmse_lkw_R2

## [1] 0.09740811

1 skill_rmse_bus_R2 <- (rmse_bus_R2_bench - rmse_bus_R2_model) / rmse_bus_R2_bench
2 skill_rmse_bus_R2

## [1] 0.1409352
```

Skill score: - PKW (ETS-ANM): Das Modell schneidet 3,234 % schlechter ab als die Naive-Methode. Der Unterschied ist jedoch nur gering. - LKW (ETS-MNM): Das Modell erzielt eine 9,741 % bessere Prognosegüte als die Mean-Methode. Der Vorteil ist moderat, aber erkennbar. - Busse (ETS-ANM): Das Modell ist 13,415 % besser als die Mean-Methode und damit klar das bessere Modell.

## 4.6 Fazit

Insgesamt weisen die ETS Modelle unterschiedliche Prognosegüten je nach Fahrzeugklasse auf. Bei den PKW ist die naive Benchmark weiterhin die überlegene Methode, während die ETS Modelle bei LKW und Bussen zu einer verbesserten Prognosegüte führen. Die Residuenanalyse zeigt allerdings, dass sowohl die Auswirkung von Feiertagen, als auch wöchentliche Muster nicht vollständig erfasst werden und zum Teil Autokorrelation vorhanden bleibt. Da ETS Modelle ausschließlich auf exponentieller Glättung basieren und keine autoregressiven oder gleitenden Abhängigkeiten berücksichtigen, werden im nächsten Schritt ARIMA Modelle aufgestellt, die eine Modellierung der Autokorrelation ermöglichen und sich somit besser eignen, um auch die Wochensaison zu erfassen. Die folgende Tabelle fasst die wichtigsten Parameter der besten ETS-Modelle je Fahrzeugklasse zusammen, inklusive der geschätzten Glättungsparameter und des Skill-Scores im Vergleich zur jeweiligen Benchmark-Methode.

```
1 # Alpha Werte von ETS Modellen
2 alpha_pkw <- fit_pkw_R2 %>% select(ANM) %>% tidy() %>% filter(term == "alpha") %>% pull(e)
3 alpha_lkw <- fit_lkw_R2 %>% select(MNM) %>% tidy() %>% filter(term == "alpha") %>% pull(e)
4 alpha_bus <- fit_bus_R2 %>% select(ANM) %>% tidy() %>% filter(term == "alpha") %>% pull(e)
5
6 gamma_pkw <- fit_pkw_R2 %>% select(ANM) %>% tidy() %>% filter(term == "gamma") %>% pull(e)
7 gamma_lkw <- fit_lkw_R2 %>% select(MNM) %>% tidy() %>% filter(term == "gamma") %>% pull(e)
8 gamma_bus <- fit_bus_R2 %>% select(ANM) %>% tidy() %>% filter(term == "gamma") %>% pull(e)
9
10 alpha_values <- c(alpha_pkw, alpha_lkw, alpha_bus)
11 gamma_values <- c(gamma_pkw, gamma_lkw, gamma_bus)
12 fahrzeugklasse <- c("PKW", "LKW", "Bus")
```

```
13 ets_model <- c("ANM", "MNM", "ANM")
14 benchmark <- c("SNaive", "Mean", "Mean")
15
16 # Tabelle
17 ets_summary <- tibble(
18   Fahrzeugklasse = fahrzeugklasse,
19   ETS_Modell = ets_model,
20   Alpha = round(alpha_values, 3), # Level
21   # kein betta, da kein Trend modelliert
22   Gamma = round(gamma_values, 3), # Saison
23   Benchmark = benchmark,
24   Skill_Score = round(c(skill_rmse_pkw_R2, skill_rmse_lkw_R2, skill_rmse_bus_R2), 3)
25 )
26
27 ets_summary
```

```
## # A tibble: 3 x 6
##   Fahrzeugklasse ETS_Modell Alpha Gamma Benchmark Skill_Score
##   <chr>          <chr>    <dbl> <dbl> <chr>         <dbl>
## 1 PKW           ANM      0.337 0.137 SNaive        -0.022
## 2 LKW           MNM      0.147 0      Mean          0.097
## 3 Bus           ANM      0.169 0      Mean          0.141
```

## 5 ARIMA

### 5.1 Theretische Überlegungen

Bei den ETS Modellen ist besonders aufgefallen, dass die Wochensaison und die Feiertage nicht erfasst worden sind. Da ARIMA explizit Autokorrelationen modelliert sind diese wahrscheinlich besser geeignet um diese Effekte zu erfassen.

Erwartungen an die Fahrzeugklassen: - PKW: In der Zeitreihe liegen starke Schwankungen an Wochenenden und Feiertagen vor, sowie eine hohe Varianz an Werktagen. Das ARIMA Modell sollte die Autokorrelation über die Woche erfassen (Feiertage bleiben wahrscheinlich weiterhin problematisch -> ARIMAX?) - LKW: die Zeitreihe weist weniger Varianz und regelmäßige Muster auf. Ein ARIMA Modell sollte die struktur gut erkennen. - Busse: Bei Bussen ist ebenfalls eine starke Wochensaison zu erkennen, die erfasst werden sollte. Feiertage sind auch zu erkennen und könnten weiterhin problematisch bleiben. SARIMA? -> Wochensaison

### 5.2 Stationarität (Bedingungen erfüllt?)

#### 5.2.1 KPSS Test (1. Durchlauf)

Stationarität mit KPSS Test prüfen. Nullhypothese: Daten sind stationär(!)

```
1 # Trainingsdaten
2
3 # PKW R2
4 train_pkw_R2 %>% features(Pkw_Gruppe_R2, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1     0.901     0.01
```

```
1 # LKW R2
2 train_lkw_R2 %>% features(Lkw_mA_R2, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1     0.146     0.1
```

```
1 # Bus R2
2 train_bus_R2 %>% features(Busse_R2, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1     1.18     0.01
```

Alle drei Zeitreihen der Fahrzeugklassen weisen einen Kpss value von 0.05 auf, daher wird die Nullhypothese wiederlegt. Die Zeitreihen sind nicht stationär, was durch die starke Wochensaison auch plausibel ist. Daher werden die Daten im Folgenden differenziert.

### 5.2.2 Differenzierung (1. Durchlauf)

```
1 # Saisonale Differenzierung
2
3 # PKW R2
4 diff_pkw <- pkw_R2_ts %>%
5   mutate(diff = difference(Pkw_Gruppe_R2, lag = 7))
6
7 # LKW R2
8 diff_lkw <- lkw_R2_ts %>%
9   mutate(diff = difference(Lkw_mA_R2, lag = 7, differences = 2))
10
11 # Bus R2
12 diff_bus <- bus_R2_ts %>%
13   mutate(diff = difference(Busse_R2, lag = 7))
```

### 5.2.3 KPSS Test (2. Durchlauf)

```
1 # PKW R2
2 diff_pkw %>% features(Pkw_Gruppe_R2, unitroot_kpss)
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>     <dbl>
## 1     0.836     0.01
```

```
1 # LKW R2
2 diff_lkw %>% features(Lkw_mA_R2, unitroot_kpss)

## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>      <dbl>
## 1     0.164        0.1
```

```
1 # Bus R2
2 diff_bus %>% features(Busse_R2, unitroot_kpss)

## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>      <dbl>
## 1     0.446        0.0572
```

Bei den PKW liegt der kpss value mit 0.01 weiterhin unter 0.05 und muss deshalb weiter differenziert werden. Bei LKW und Bussen liegt der kpss value über 0.05. Die Stationarität wurde bei den beiden erreicht.

#### 5.2.4 Differenzierung (2. Durchlauf)

```
1 # PKW R2
2 diff2_pkw <- diff_pkw %>%
3   mutate(diff2 = difference(diff, lag = 1))
```

#### 5.2.5 KPSS Test (3. Durchlauf)

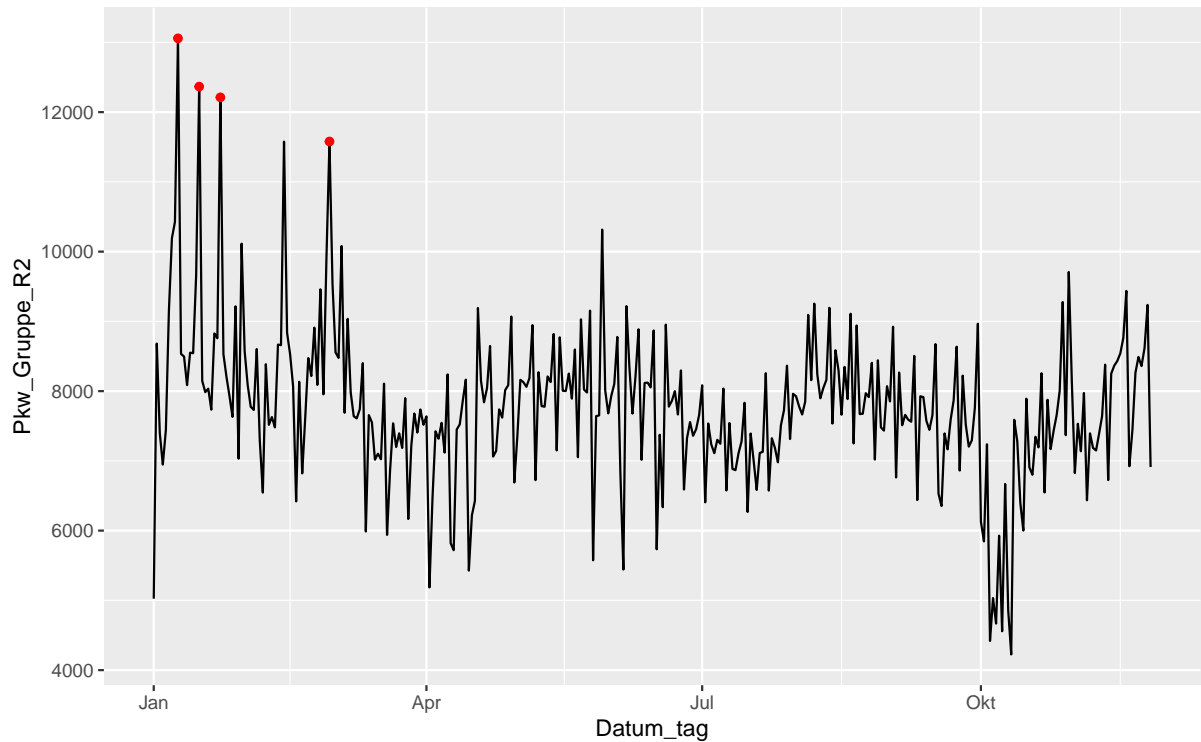
```
1 # PKW R2
2 diff2_pkw %>% features(diff2, unitroot_kpss)

## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##   <dbl>      <dbl>
## 1     0.0622        0.1
```

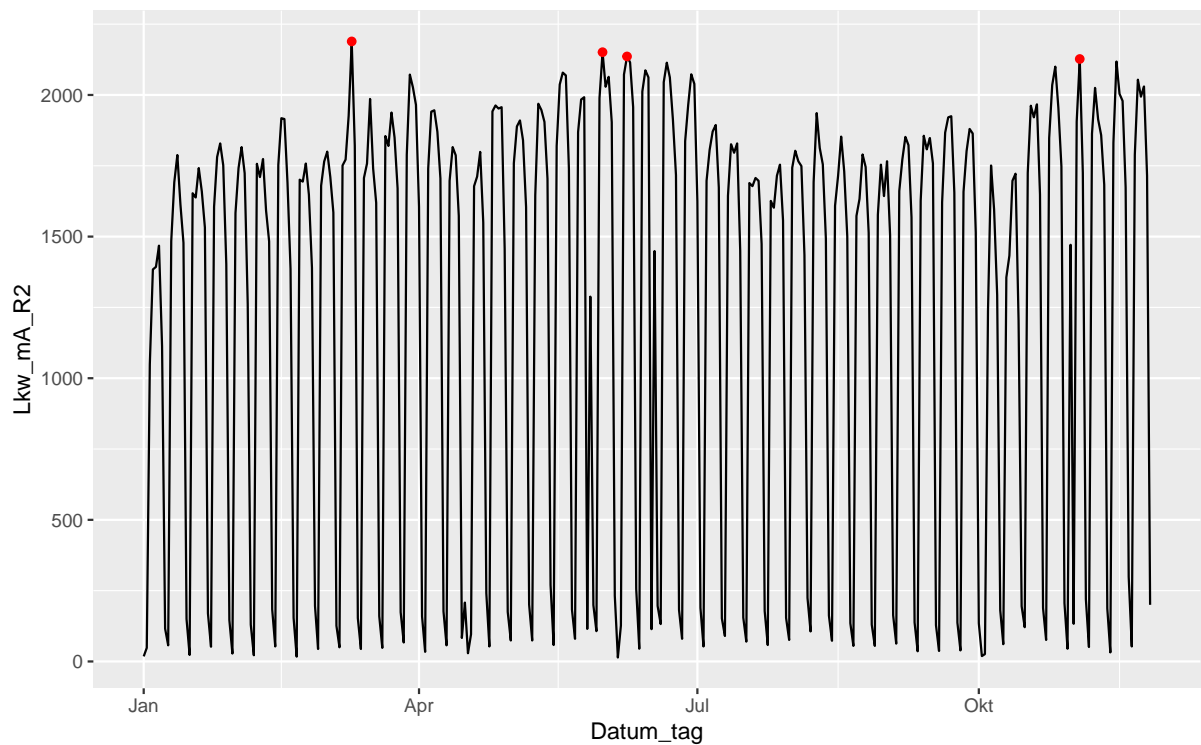
Alle Zeitreihen sind jetzt stationär. LKW und Busse direkt nach saisonaler Differenzierung. PKW eine Differenzierung danach.

#### 5.2.6 Pürung auf Extremwerte

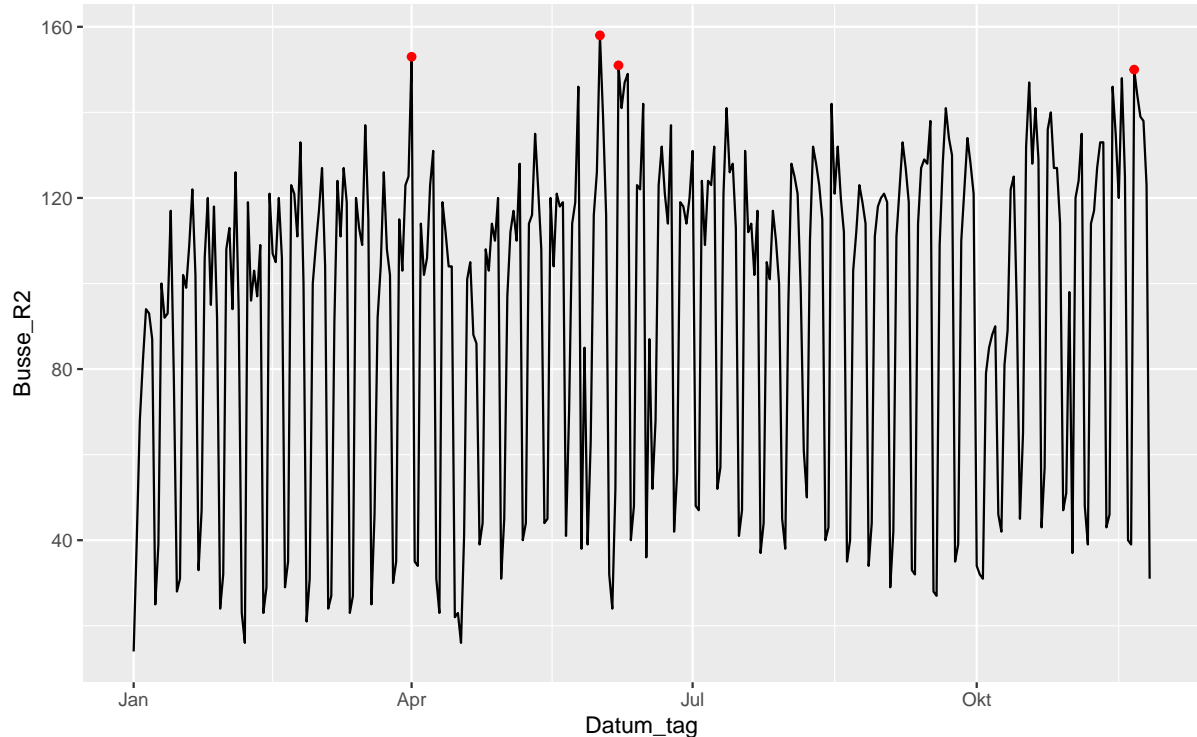
```
1 #library(ggplot2)
2 ggplot(train_pkw_R2, aes(x = Datum_tag, y = Pkw_Gruppe_R2)) +
3   geom_line() +
4   geom_point(data = subset(train_pkw_R2, Pkw_Gruppe_R2 > quantile(Pkw_Gruppe_R2, 0.99)),
5             aes(x = Datum_tag, y = Pkw_Gruppe_R2), color = "red")
```



```
1 ggplot(train_lkw_R2, aes(x = Datum_tag, y = Lkw_mA_R2)) +  
2   geom_line() +  
3   geom_point(data = subset(train_lkw_R2, Lkw_mA_R2 > quantile(Lkw_mA_R2, 0.99)),  
4             aes(x = Datum_tag, y = Lkw_mA_R2), color = "red")
```



```
1 ggplot(train_bus_R2, aes(x = Datum_tag, y = Busse_R2)) +  
2   geom_line() +  
3   geom_point(data = subset(train_bus_R2, Busse_R2 > quantile(Busse_R2, 0.99)),  
4             aes(x = Datum_tag, y = Busse_R2), color = "red")
```



Es sind einzelne extremere Werte zu Beginn der Zeitreihe vorhanden, Werte die außerhalb des 99% Quantils liegen. Da die Werte in das Muster zu Beginn der Zeitreihe passen und ARIMA Modelle außerdem relativ robust gegenüber Ausreißern sind werden die Werte so beibehalten und nicht weiter behandelt. Die 4 extremsten Werte bei den LKW und Bussen sind noch weniger auffällig.

## 5.3 Praxis

### 5.3.1 Modelle

```
1 # PKW R2  
2 fit_pkw_arima <- train_pkw_R2 %>%  
3   model(arima = ARIMA(Pkw_Gruppe_R2))  
4 report(fit_pkw_arima)
```

```
## Series: Pkw_Gruppe_R2  
## Model: ARIMA(0,1,1)(2,0,0)[7]  
##  
## Coefficients:  
##          ma1      sar1      sar2  
##      -0.6197  0.3181  0.2957  
## s.e.   0.0556  0.0551  0.0563  
##
```



```
## sigma^2 estimated as 792213: log likelihood=-2701.33
## AIC=5410.66 AICc=5410.78 BIC=5425.84
```

```
1 # LKW R2
2 fit_lkw_arima <- train_lkw_R2 %>%
3   model(arima = ARIMA(Lkw_mA_R2))
4 report(fit_lkw_arima)
```

```
## Series: Lkw_mA_R2
## Model: ARIMA(1,0,0)(1,1,0)[7] w/ drift
##
## Coefficients:
##          ar1      sar1  constant
##      0.2407 -0.4727    9.9730
## s.e. 0.0539  0.0483   18.8853
##
## sigma^2 estimated as 114886: log likelihood=-2339.47
## AIC=4686.95 AICc=4687.07 BIC=4702.06
```

```
1 # Bus R2
2 fit_bus_arima <- train_bus_R2 %>%
3   model(arima = ARIMA(Busse_R2))
4 report(fit_bus_arima)
```

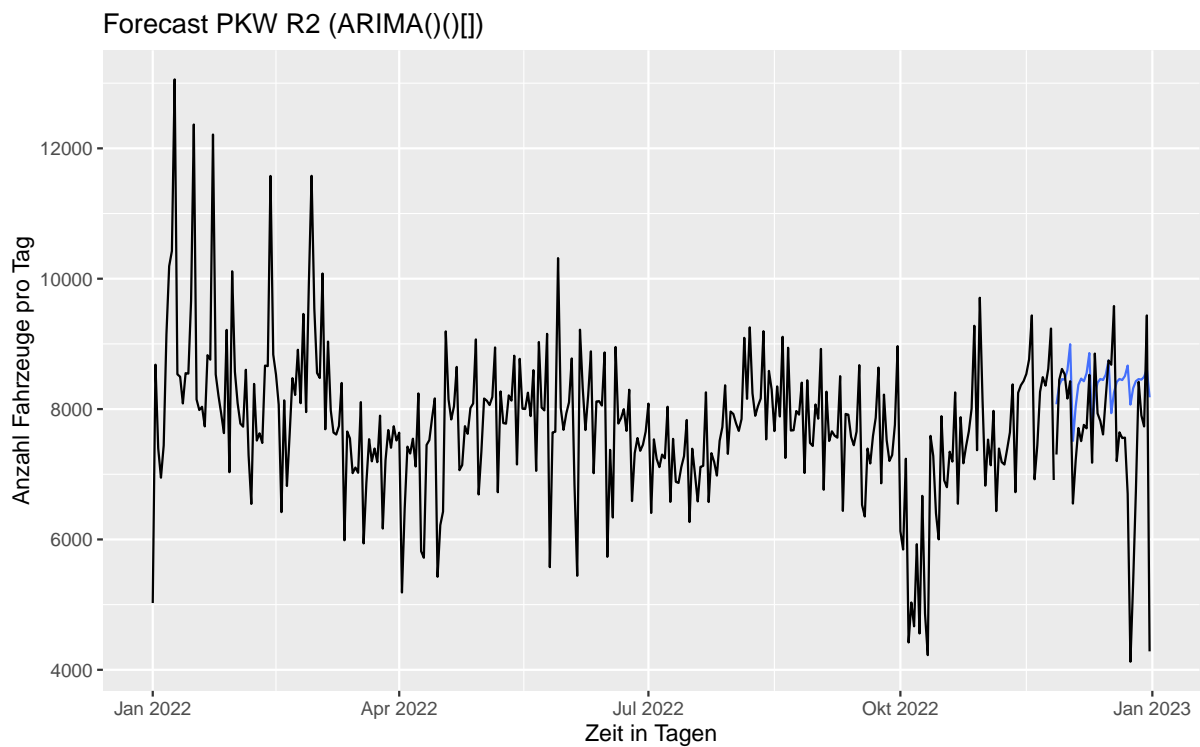
```
## Series: Busse_R2
## Model: ARIMA(1,0,1)(2,1,2)[7]
##
## Coefficients:
##          ar1      ma1      sar1      sar2      sma1      sma2
##      0.6242 -0.3518 -0.6975 -0.0463 -0.0749 -0.7387
## s.e. 0.1416  0.1697  0.1421  0.0836  0.1370  0.1368
##
## sigma^2 estimated as 291.2: log likelihood=-1377.68
## AIC=2769.36 AICc=2769.72 BIC=2795.81
```

### 5.3.2 Forecasts

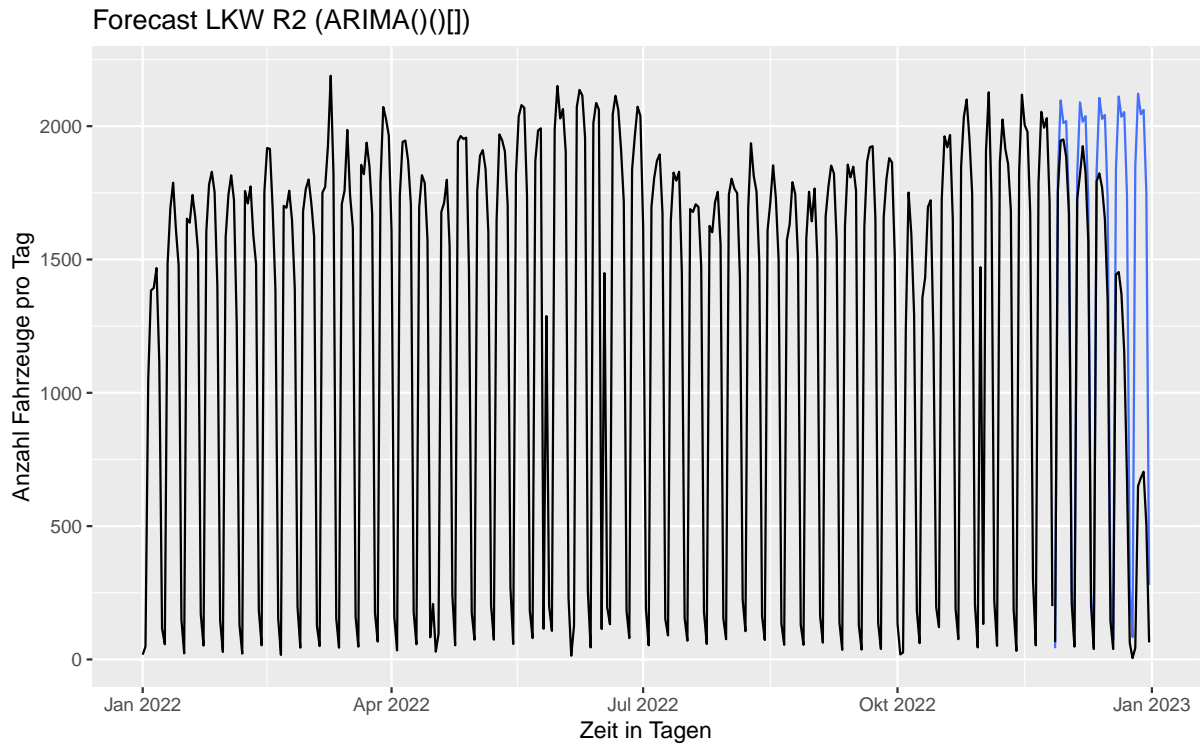
```
1 fc_pkw_arima <- fit_pkw_arima %>%
2   forecast(new_data = test_pkw_R2)
3
4 fc_lkw_arima <- fit_lkw_arima %>%
5   forecast(new_data = test_lkw_R2)
6
7 fc_bus_arima <- fit_bus_arima %>%
8   forecast(new_data = test_bus_R2)
```

### 5.3.3 Plots

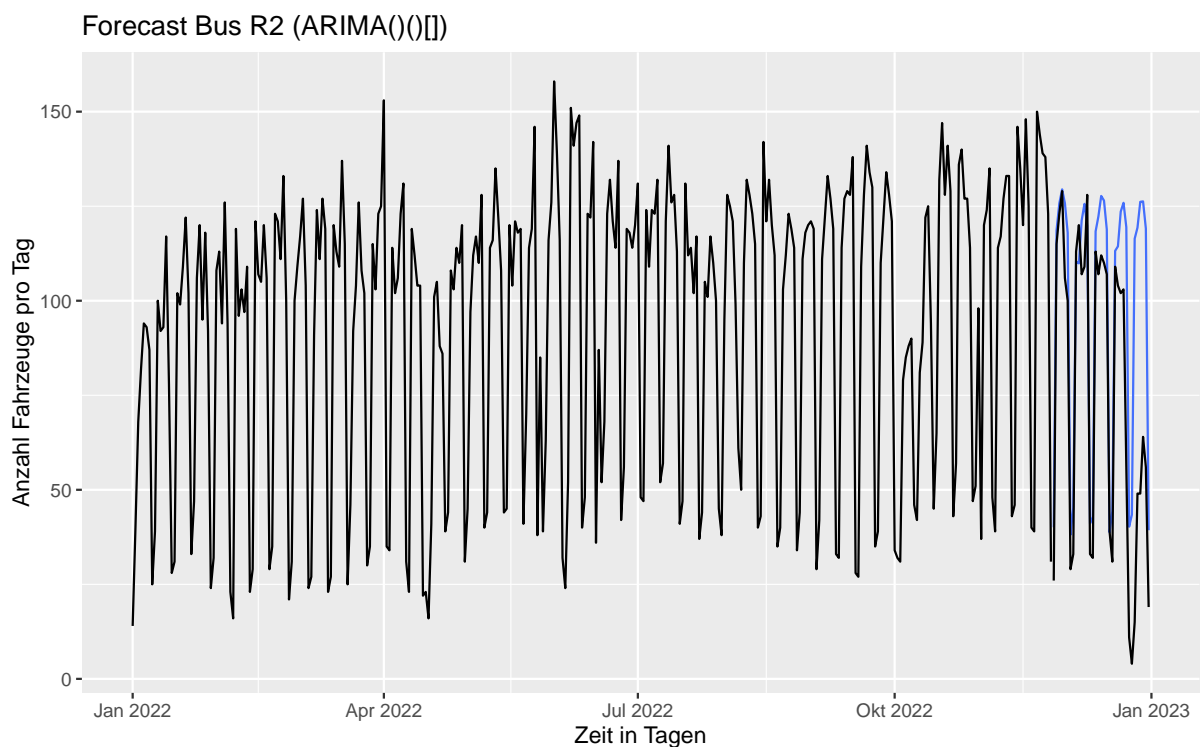
```
1 fc_pkw_arima %>%
2   autoplot(train_pkw_R2, level = NULL) +
3   autolayer(test_pkw_R2, Pkw_Gruppe_R2, color = "black") +
4   labs(title = "Forecast PKW R2 (ARIMA())()",
5         y = "Anzahl Fahrzeuge pro Tag",
6         x = "Zeit in Tagen") +
7   guides(color = guide_legend(title = "Modell"))
```



```
1 fc_lkw_arima %>%
2   autoplot(train_lkw_R2, level = NULL) +
3   autolayer(test_lkw_R2, Lkw_mA_R2, color = "black") +
4   labs(title = "Forecast LKW R2 (ARIMA())()",
5         y = "Anzahl Fahrzeuge pro Tag",
6         x = "Zeit in Tagen") +
7   guides(color = guide_legend(title = "Modell"))
```



```
1 fc_bus_arima %>%  
2   autoplot(train_bus_R2, level = NULL) +  
3   autolayer(test_bus_R2, Busse_R2, color = "black") +  
4   labs(title = "Forecast Bus R2 (ARIMA()())",  
5        y = "Anzahl Fahrzeuge pro Tag",  
6        x = "Zeit in Tagen") +  
7   guides(color = guide_legend(title = "Modell"))
```



## 5.4 Bewertung der Modelgüte

Heteroskedastisch Normalverteilt Um Null verteilt

### 5.4.1 AIC, AICc und BIC

```
1 report(fit_pkw_arima)
```

```
## Series: Pkw_Gruppe_R2
## Model: ARIMA(0,1,1)(2,0,0)[7]
##
## Coefficients:
##          ma1      sar1      sar2
##        -0.6197  0.3181  0.2957
## s.e.    0.0556  0.0551  0.0563
##
## sigma^2 estimated as 792213:  log likelihood=-2701.33
## AIC=5410.66   AICc=5410.78   BIC=5425.84
```

```
1 report(fit_lkw_arima)
```

```
## Series: Lkw_mA_R2
## Model: ARIMA(1,0,0)(1,1,0)[7] w/ drift
##
## Coefficients:
##          ar1      sar1  constant
##         0.2407 -0.4727    9.9730
## s.e.    0.0539  0.0483   18.8853
##
## sigma^2 estimated as 114886:  log likelihood=-2339.47
## AIC=4686.95   AICc=4687.07   BIC=4702.06
```

```
1 report(fit_bus_arima)
```

```
## Series: Busse_R2
## Model: ARIMA(1,0,1)(2,1,2)[7]
##
## Coefficients:
##          ar1      ma1      sar1      sar2      sma1      sma2
##         0.6242 -0.3518 -0.6975 -0.0463 -0.0749 -0.7387
## s.e.    0.1416  0.1697  0.1421  0.0836  0.1370  0.1368
##
## sigma^2 estimated as 291.2:  log likelihood=-1377.68
## AIC=2769.36   AICc=2769.72   BIC=2795.81
```

AIC, BIC liegen bei allen nah beieinander.

### 5.4.2 MAE und RMSE

```

1 # In-sample accuracy (Trainingsdaten)
2 acc_pkw_R2_arma_train <- accuracy(fit_pkw_arma)
3 acc_lkw_R2_arma_train <- accuracy(fit_lkw_arma)
4 acc_bus_R2_arma_train <- accuracy(fit_bus_arma)
5
6 # Out-of sample accuracy (Forecast, Testdaten)
7 acc_pkw_R2_arma_test <- accuracy(fc_pkw_arma, test_pkw_R2)
8 acc_lkw_R2_arma_test <- accuracy(fc_lkw_arma, test_lkw_R2)
9 acc_bus_R2_arma_test <- accuracy(fc_bus_arma, test_bus_R2)
10
11 list(acc_pkw_R2_arma_train, acc_lkw_R2_arma_train, acc_bus_R2_arma_train, acc_pkw_R2_arma_test, acc_lkw_R2_arma_test, acc_bus_R2_arma_test)

```

```

## [[1]]
## # A tibble: 1 x 10
##   .model .type      ME  RMSE  MAE   MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arma  Training  3.88  885.  613. -0.928  8.09  0.853  0.784  0.0617
##
## [[2]]
## # A tibble: 1 x 10
##   .model .type      ME  RMSE  MAE   MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arma  Training  1.37  334.  153. -54.3  68.1  0.929  0.847  0.000175
##
## [[3]]
## # A tibble: 1 x 10
##   .model .type      ME  RMSE  MAE   MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arma  Training  2.23  16.7  11.3 -3.15  17.2  0.788  0.725 -0.00752
##
## [[4]]
## # A tibble: 1 x 10
##   .model .type      ME  RMSE  MAE   MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arma  Test    -734. 1326.  954. -13.0  15.4   NaN    NaN  0.324
##
## [[5]]
## # A tibble: 1 x 10
##   .model .type      ME  RMSE  MAE   MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arma  Test    -409.  642.  410. -231.  233.   NaN    NaN  0.666
##
## [[6]]
## # A tibble: 1 x 10
##   .model .type      ME  RMSE  MAE   MPE  MAPE  MASE  RMSSE  ACF1
##   <chr>  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arma  Test    -22.1  33.9  23.3 -87.2  88.2   NaN    NaN  0.719

```

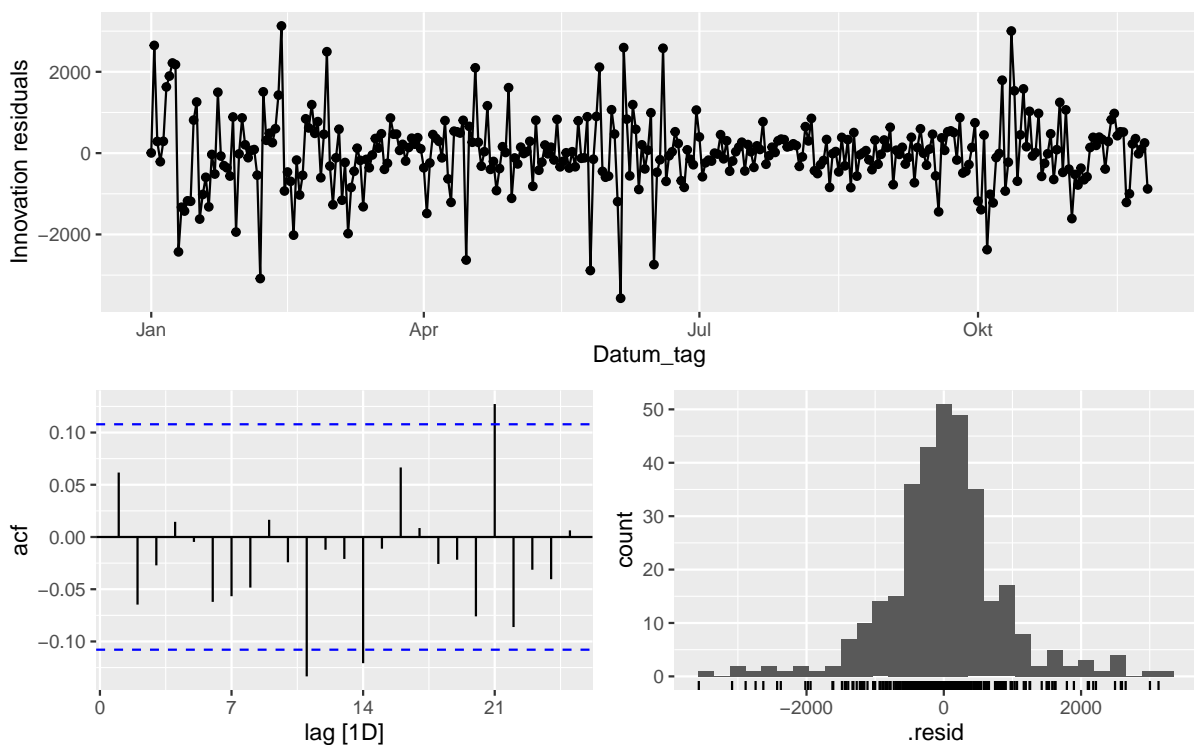
- PKW: noch deutlich unterprognostiziert (ME: -734), MAE RMSE ebenfalls relativ groß -> hoher Fehler, leichte Autokorrelation (SARIMA wahrscheinlich besser) (ca. 12000 PKW pro Tag)

- LKW: ebenfalls unterprognostiziert, aber weniger, Fehler ebenfalls geringer, Reisdien korrelieren stark (ACF: 0.666) (ca. 2000 LKW pro Tag)
- Bus: unterprognostiziert noch stärkere Autokorrelation

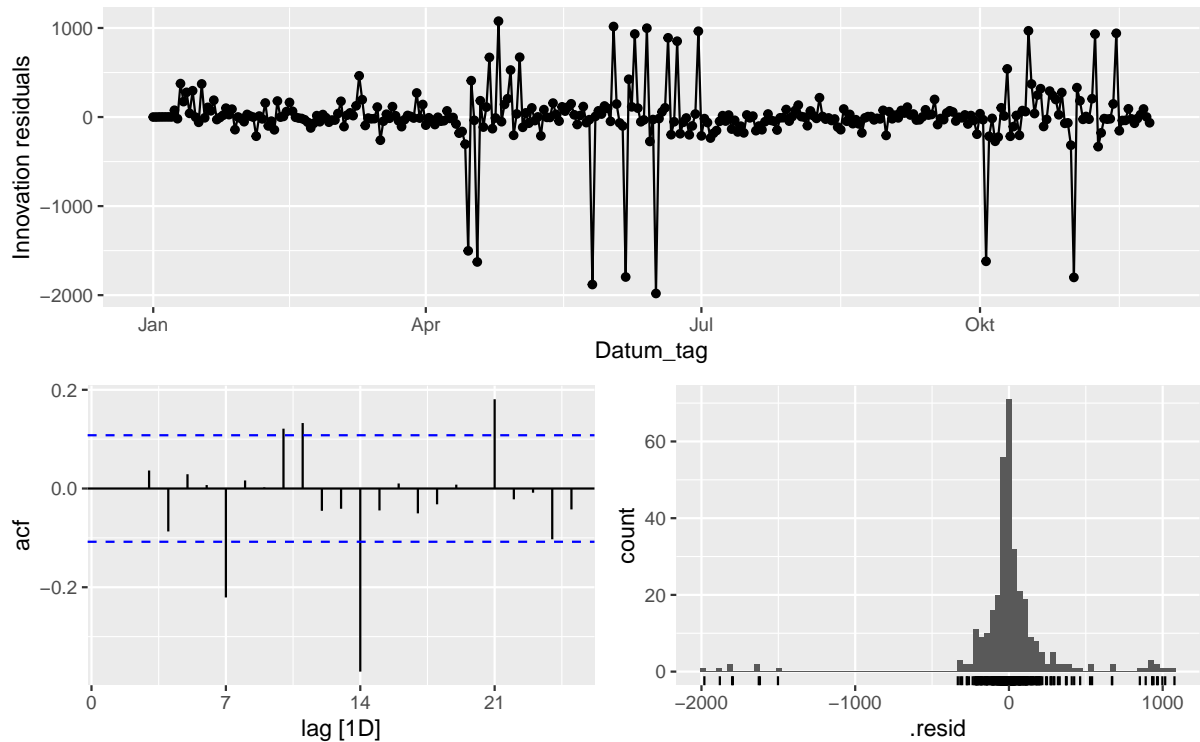
ACF -> Autokorrelation (0.1, 0.3, 0.5 +/-) Bei allen Autokorrelation vorhanden -> Strukturen werden nicht erfasst -> SARIMA

### 5.4.3 Residuenanalyse mit Zeit, ACF, Hist

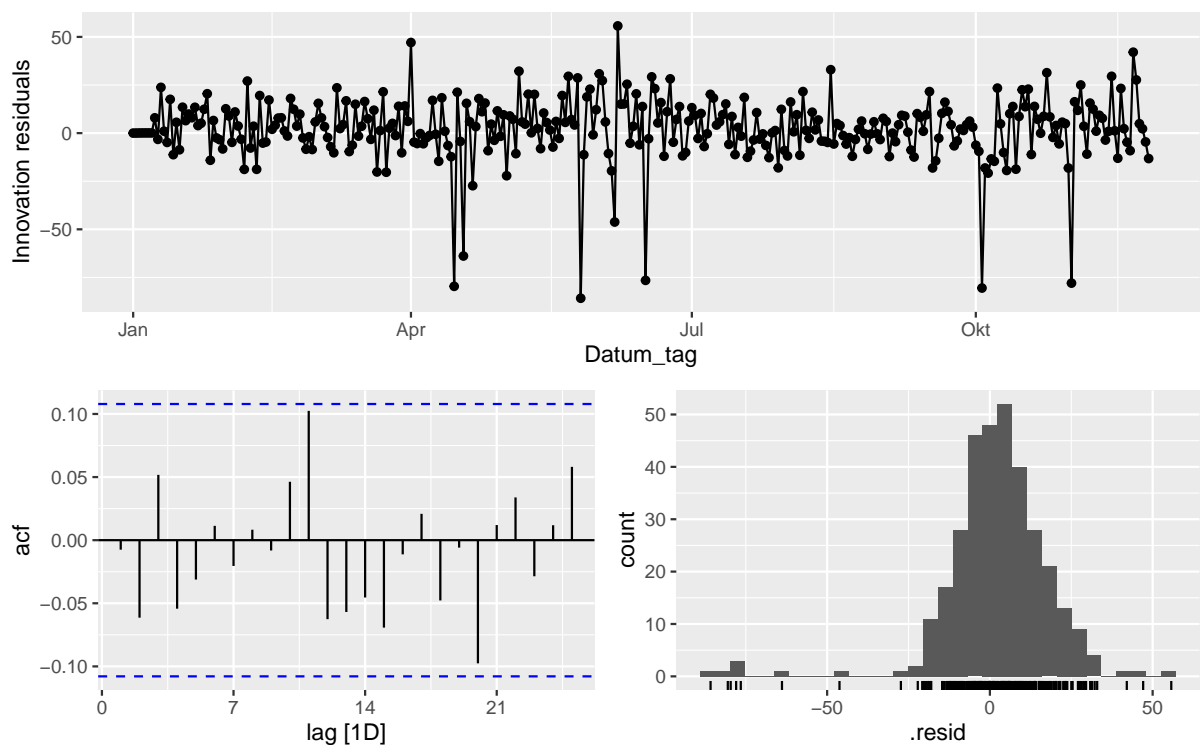
```
1 fit_pkw_arima %>% select(arima) %>% gg_tsresiduals()
```



```
1 fit_lkw_arima %>% select(arima) %>% gg_tsresiduals()
```



```
1 fit_bus_arima %>% select(arima) %>% gg_tsresiduals()
```

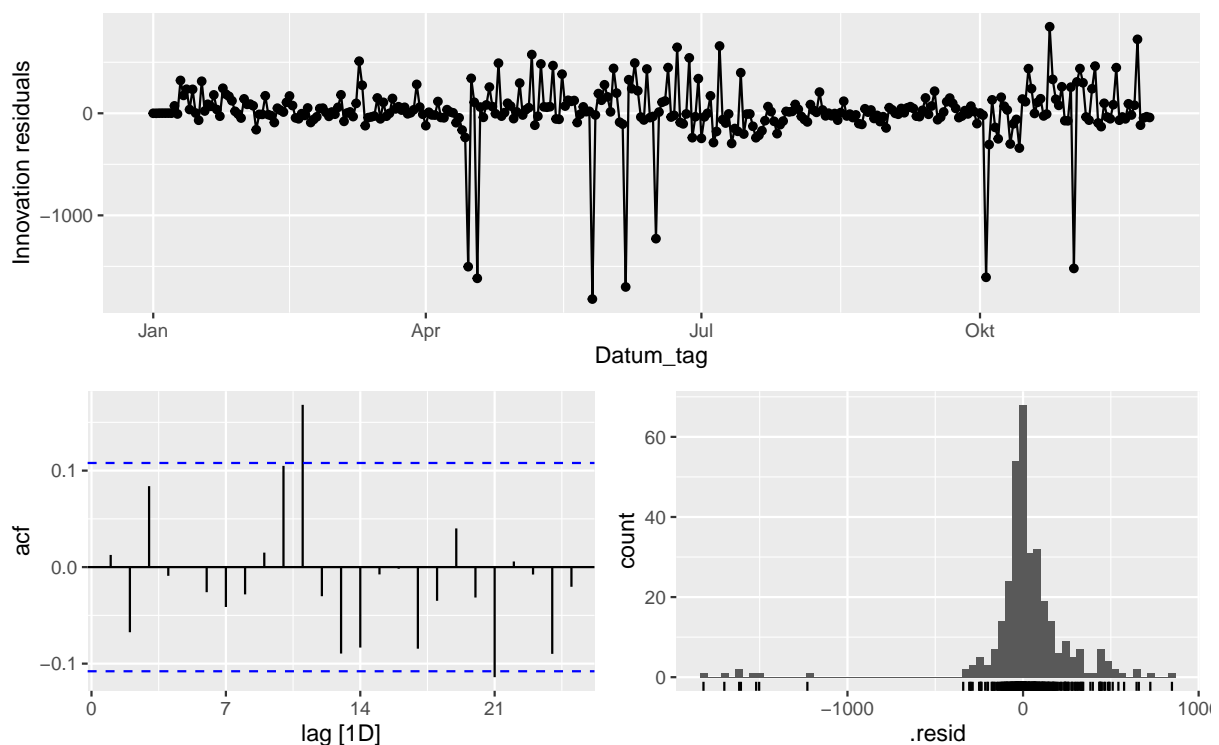


PKW: Streuung (-2000 bis 2000) da bei Zeitreihe der Residuen aber am Anfang mehr als am Ende; ACF Werte über Linie: 11,14,21, Histogramm gut normalverteilt LKW: Streuung gering (einzelnen Werte extrem), ACF: Lags deutlich bei 7 noch stärker bei 14, ganz leicht drüber bei 10, 11, etwas weiter drüber bei 21; Histogramm an sich normalverteilt um null, Busse: streuung okay, wenige nach unten bei -70 (Feiertage wie bei PKW und LKW auch), ACF: keine Lags über Grenzne, Histogramm: normalverteilt um 0, wenige werte bei -100 und noch weniger bei 50

```
1 # LKW R2
2 fit_lkw_arma_man <- train_lkw_R2 %>%
3   model(arma = ARIMA(Lkw_mA_R2 ~ pdq(1,0,1) + PDQ(3,1,0))) #101 310
4 report(fit_lkw_arma_man)
```

```
## Series: Lkw_mA_R2
## Model: ARIMA(1,0,1)(3,1,0)[7]
##
## Coefficients:
##      ar1      ma1      sar1      sar2      sar3
##      0.6366 -0.4630 -0.8177 -0.6299 -0.2418
## s.e.  0.2012  0.2316  0.0581  0.0657  0.0585
##
## sigma^2 estimated as 86306:  log likelihood=-2294.58
## AIC=4601.15   AICc=4601.42   BIC=4623.82
```

```
1 fit_lkw_arma_man %>% select(arma) %>% gg_tsresiduals()
```



Ausschlag bei ACF Plot bei 11 bleibt, aber Wochensaison wurde erfasst, Normalverteilung der Residuen ist vorhanden, wenn auch weiterhin durch Extremwerte im negativen Bereich durch Feiertage (gleiches Problem bei Zeitreihe der Residuen)

```
1 # Forecast LKW ARIMA (manuell)
2 fc_lkw_arma_man <- fit_lkw_arma_man %>%
3   forecast(new_data = test_lkw_R2)
4
5 acc_lkw_R2_arma_man_test <- accuracy(fc_lkw_arma_man, test_lkw_R2)
6 acc_lkw_R2_arma_man_test
```



```
## # A tibble: 1 x 10
##   .model .type    ME  RMSE   MAE   MPE   MAPE   MASE  RMSSE  ACF1
##   <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima  Test  -336.  590.  358. -191.  194.   NaN   NaN  0.645
```

#### 5.4.4 Ljung Box Test

```
1 # Residuen aus Modellen
2 res_pkw <- augment(fit_pkw_arma)
3 res_lkw_auto <- augment(fit_lkw_arma)
4 res_lkw_man <- augment(fit_lkw_arma_man)
5 res_bus <- augment(fit_bus_arma)
6
7
8 # PKW
9 Box.test(res_pkw$.resid, type = "Ljung-Box", lag = 20)

##
## Box-Ljung test
##
## data:  res_pkw$.resid
## X-squared = 21.908, df = 20, p-value = 0.3455

1 # LKW
2 Box.test(res_lkw_auto$.resid, type = "Ljung-Box", lag = 20)

##
## Box-Ljung test
##
## data:  res_lkw_auto$.resid
## X-squared = 81.947, df = 20, p-value = 1.83e-09

1 # LKW - manuell
2 Box.test(res_lkw_man$.resid, type = "Ljung-Box", lag = 20)

##
## Box-Ljung test
##
## data:  res_lkw_man$.resid
## X-squared = 27.945, df = 20, p-value = 0.1107

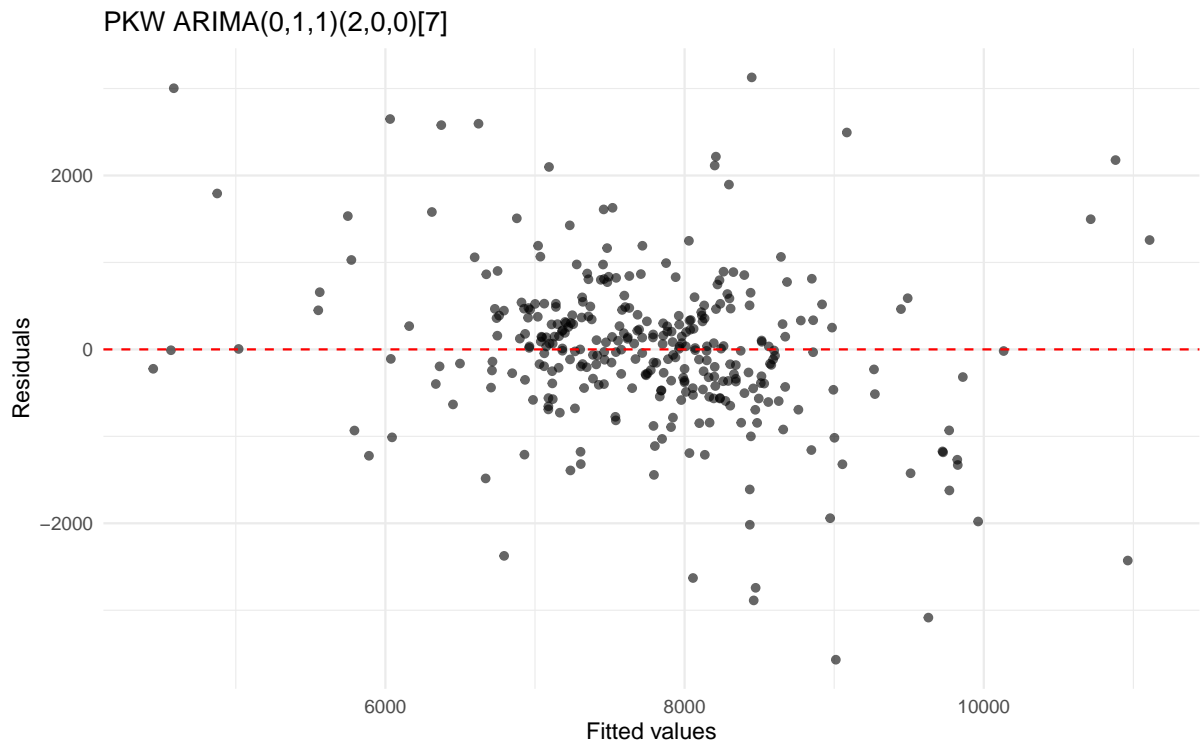
1 # Bus
2 Box.test(res_bus$.resid, type = "Ljung-Box", lag = 20)

##
## Box-Ljung test
##
## data:  res_bus$.resid
## X-squared = 17.289, df = 20, p-value = 0.6341
```

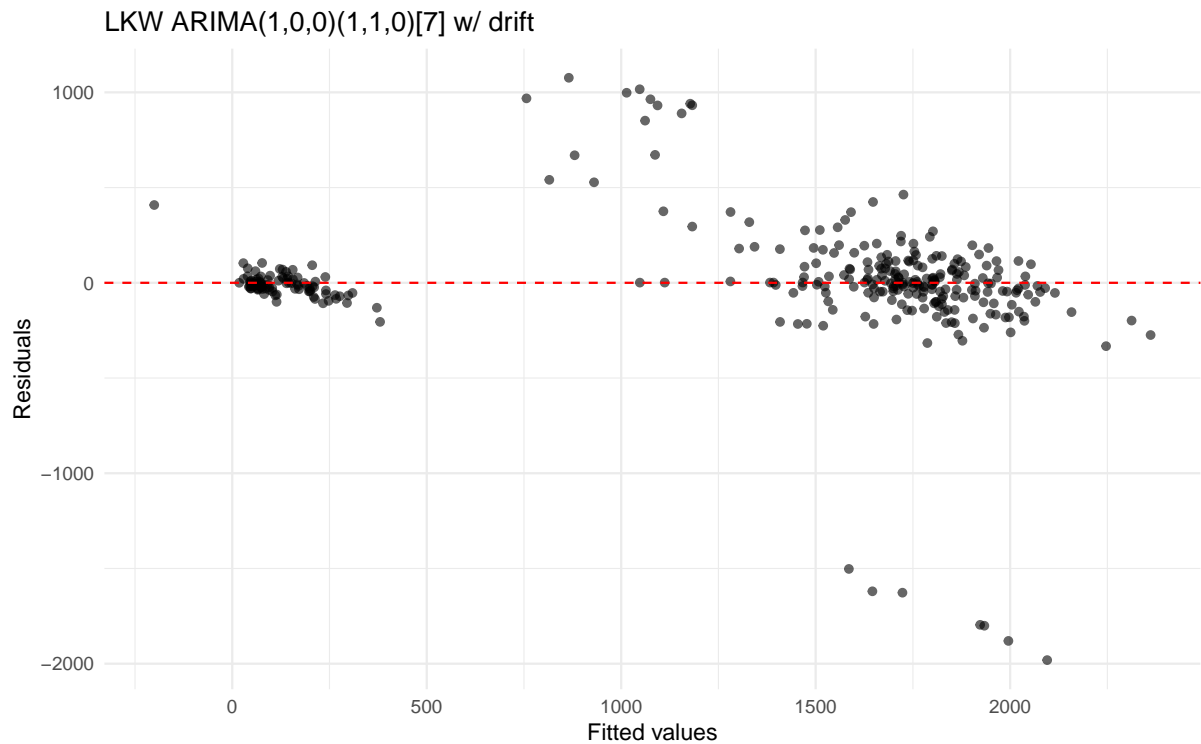
Bei den beiden automatischen Modellen für PKW und Busse, sowie beim manuellen ARIMA Modell für die LKW ist keine signifikante Restautokorrelation mehr zu erkennen. Das automatische ARIMA Modell für die LKW erfasst die Wochenstruktur nicht vollständig, das manuelle Modell hingegen, erfasst die Struktur deutlich besser (Forecast sollte stabiler sein).

#### 5.4.5 Residuen vs. fitted Values

```
1 plot_resid_vs_fitted(fit_pkw_arima %>% select(arima), "PKW ARIMA(0,1,1)(2,0,0)[7]")
```

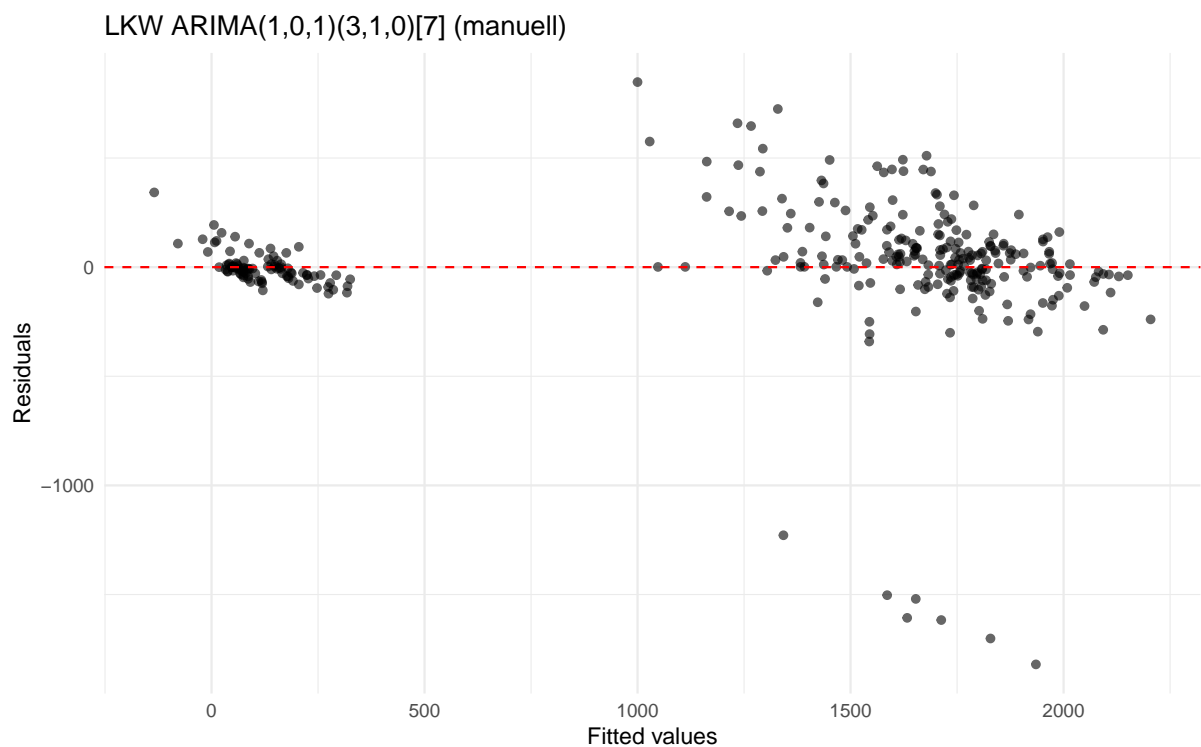


```
1 plot_resid_vs_fitted(fit_lkw_arima %>% select(arima), "LKW ARIMA(1,0,0)(1,1,0)[7] w/ drift")
```



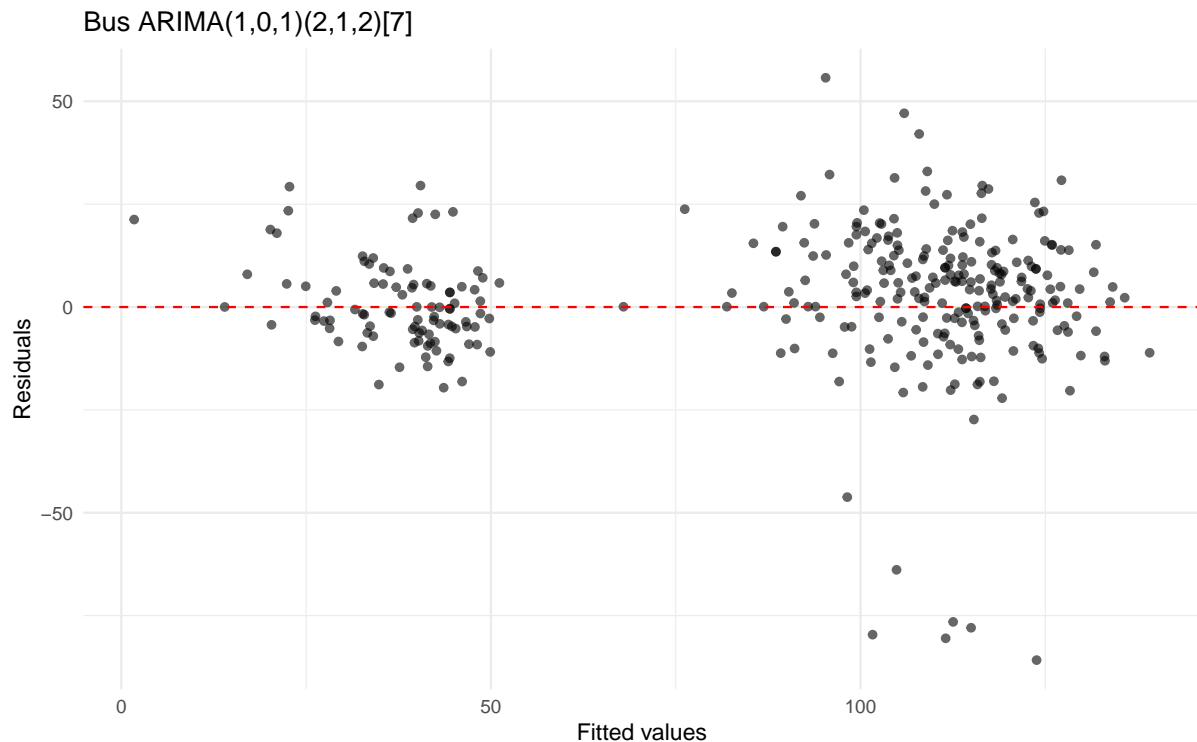
1

```
plot_resid_vs_fitted(fit_lkw_arima_man %>% select(arima), "LKW ARIMA(1,0,1)(3,1,0)[7] (ma
```



1

```
plot_resid_vs_fitted(fit_bus_arima %>% select(arima), "Bus ARIMA(1,0,1)(2,1,2)[7]")
```



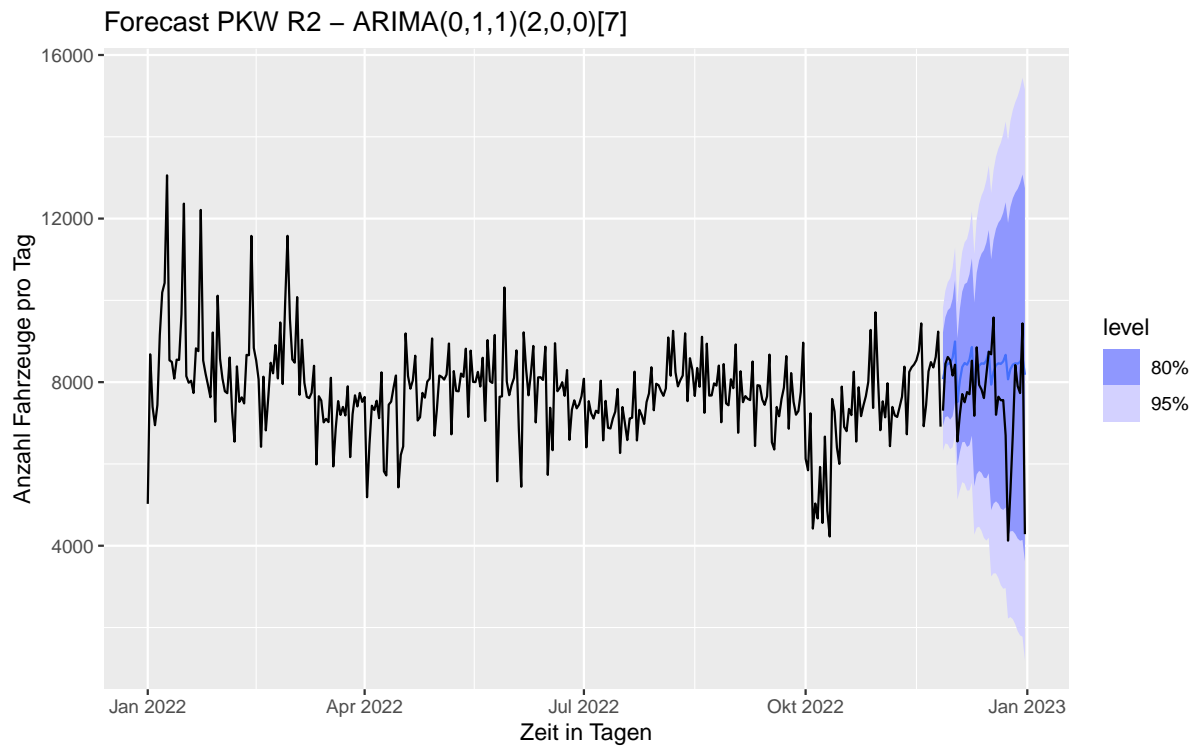
## 5.5 Erkenntnisse

- PKW ARIMA(0,1,1)(2,0,0)[7]: Residuenanalyse unauffällig (Lags 11,14,21 aber nur noch sehr leicht), normalverteilt, Ljung Box Test zeigt keine signifikante Autokorrelation mehr. Das Modell bildet die Wochensaison ab. Einzelne Ausreißer sind auf Feiertage zurückzuführen (können so vom Modell nicht erfasst werden). Insgesamt ist Modell gut für Prognose geeignet
- LKW ARIMA(1,0,0)(1,1,0)[7] w/ drift: Modell schneidet deutlich schlechter ab, signifikante Restautokorrelation (Ljung Box Test), nicht für Prognose geeignet, daher manuelles ARIMA Modell ->
- LKW ARIMA(1,0,1)(3,1,0)[7] (manuell): deutlich reduzierte Restautokorrelation, die nicht mehr signifikant ist (Ljung Box), Wochensaison wird erfasst, Residuen weitgehend unauffällig (ACF Plot: leicht über Grenze bei 11, Histogramm: einige stark negative Ausschläge, logisch durch Feiertage); Modell deutlich besser für Prognose geeignet
- Bus ARIMA(1,0,1)(2,1,2)[7]: Residuenanalyse unauffällig, keine Restautokorrelation, Wochensaison erfasst, Modell gut für Prognose geeignet

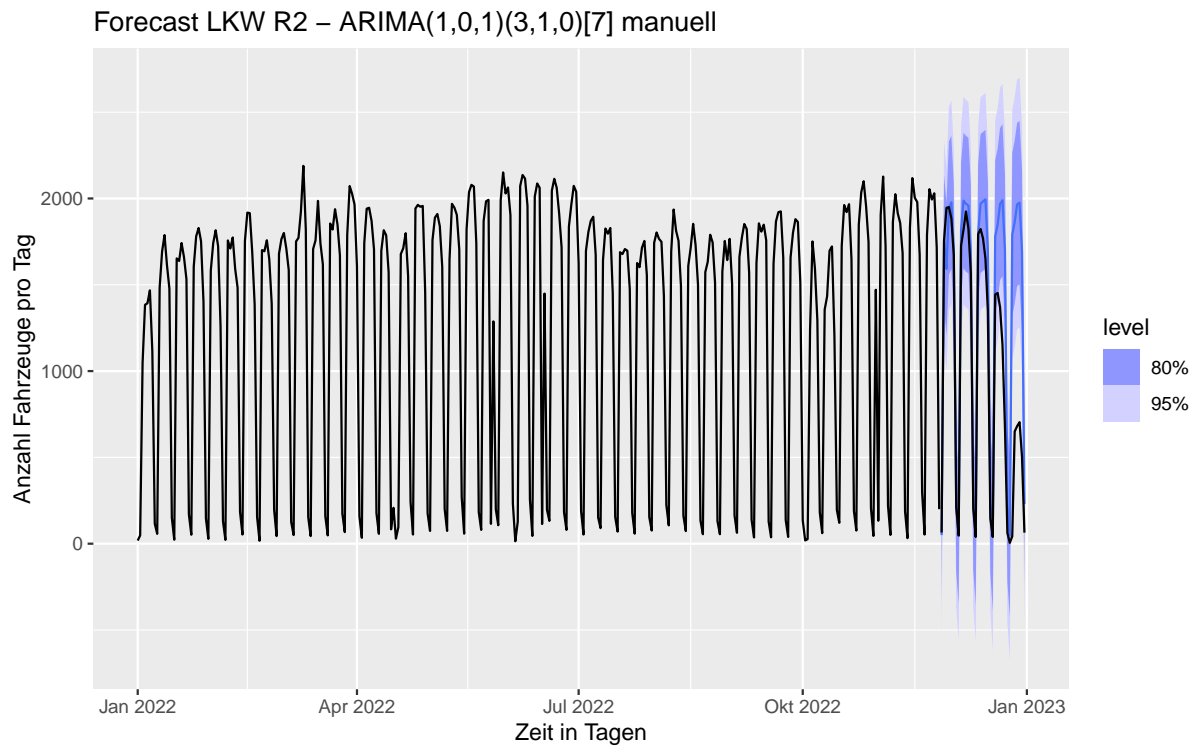
## 5.6 Beste Modelle

Forecast

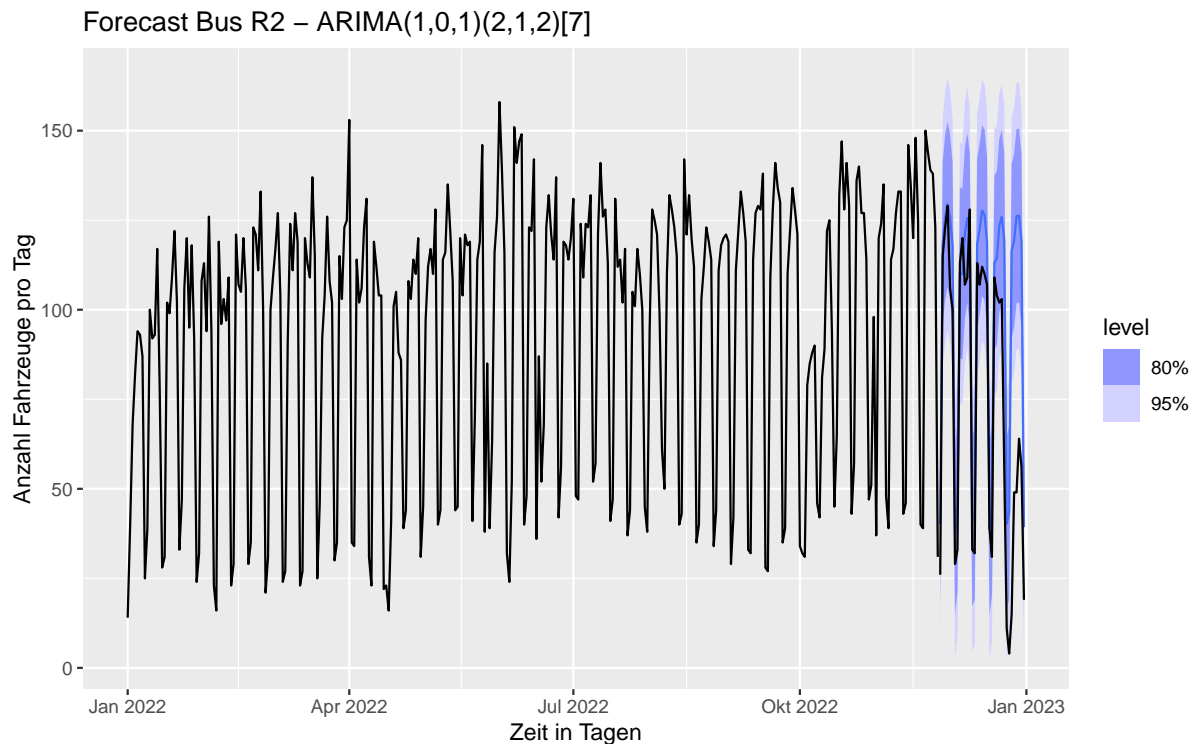
```
1 # PKW R2 - ARIMA
2 fc_pkw_arima %>%
3   autoplot(train_pkw_R2, level = c(80, 95)) +
4   autolayer(test_pkw_R2, Pkw_Gruppe_R2, color = "black") +
5   labs(title = "Forecast PKW R2 - ARIMA(0,1,1)(2,0,0)[7]",
6        y = "Anzahl Fahrzeuge pro Tag",
7        x = "Zeit in Tagen") +
8   guides(color = guide_legend(title = "Modell"))
```



```
1 # LKW R2 - ARIMA (manuell)
2 fc_lkw_arima_man %>%
3   autoplot(train_lkw_R2, level = c(80, 95)) +
4   autolayer(test_lkw_R2, Lkw_ma_R2, color = "black") +
5   labs(title = "Forecast LKW R2 - ARIMA(1,0,1)(3,1,0)[7] manuell",
6         y = "Anzahl Fahrzeuge pro Tag",
7         x = "Zeit in Tagen") +
8   guides(color = guide_legend(title = "Modell"))
```



```
1 # Bus R2 - ARIMA
2 fc_bus_arima %>%
3   autoplot(train_bus_R2, level = c(80, 95)) +
4   autolayer(test_bus_R2, Busse_R2, color = "black") +
5   labs(title = "Forecast Bus R2 - ARIMA(1,0,1)(2,1,2)[7]",
6        y = "Anzahl Fahrzeuge pro Tag",
7        x = "Zeit in Tagen") +
8   guides(color = guide_legend(title = "Modell"))
```



### 5.6.1 Vergleich von ETS Modellen und Naiven Methoden

Skill score

```
1 # CRPS für Benchmarks
2 rmse_pkw_R2_bench <- acc_pkw_R2 %>%
3   filter(.model == "snaive", .type == "Test") %>%
4   pull(RMSE)
5
6 rmse_lkw_R2_bench <- acc_lkw_R2 %>%
7   filter(.model == "mean", .type == "Test") %>%
8   pull(RMSE)
9
10 rmse_bus_R2_bench <- acc_bus_R2 %>%
11   filter(.model == "mean", .type == "Test") %>%
12   pull(RMSE)
13
14
15 # CRPS für ARIMA Modelle
16 rmse_pkw_R2_arima <- acc_pkw_R2_arima_test %>%
17   filter(.model == "arima", .type == "Test") %>%
18   pull(RMSE)
19
20 rmse_lkw_R2_arima_man <- acc_lkw_R2_arima_man_test %>%
21   filter(.model == "arima", .type == "Test") %>%
22   pull(RMSE)
23 t
```

```
## function (x)
```

```
## UseMethod("t")
## <bytecode: 0x0000011759c4e5b0>
## <environment: namespace:base>
```

```
1 rmse_bus_R2_arma <- acc_bus_R2_arma_test %>%
2   filter(.model == "arma", .type == "Test") %>%
3   pull(RMSE)
4
5
6 # Skill Score
7 skill_rmse_pkw_R2_arma <- (rmse_pkw_R2_bench - rmse_pkw_R2_arma) / rmse_pkw_R2_bench
8 skill_rmse_lkw_R2_arma_man <- (rmse_lkw_R2_bench - rmse_lkw_R2_arma_man) / rmse_lkw_R2_bench
9 skill_rmse_bus_R2_arma <- (rmse_bus_R2_bench - rmse_bus_R2_arma) / rmse_bus_R2_bench
10
11 skill_rmse_pkw_R2_arma
```

```
## [1] -0.1336793
```

```
1 skill_rmse_lkw_R2_arma_man
```

```
## [1] 0.2448024
```

```
1 skill_rmse_bus_R2_arma
```

```
## [1] 0.226202
```

Skill score: - PKW ARIMA(0,1,1)(2,0,0)[7]: Das Modell schneidet 13,367% schlechter ab, als die naive Methode. - LKW ARIMA(1,0,1)(3,1,0)[7] (manuell): Das Modell schneidet 24,480% besser ab, als die naive Methode und damit nochmal deutlich besser als das ETS(MNM) Modell. - Bus ARIMA(1,0,1)(2,1,2)[7]: Das Modell schneidet 22,620% besser ab, als die naive Methode und damit auch nochmal besser als das ETS(ANM) Modell

## 5.7 Fazit

Insgesamt erfassen die ARIMA Modelle die Autokorrelationen besser als die ETS Modelle. Die Wochensaison wird erkannt und verbessert die Prognosegüte deutlich. Feiertage stellen aber weiterhin ein Problem dar, insbesondere die Weihnachtszeit im Testzeitraum. Die Prognosegüte konnte bei LKW und Bussen verbessert werden, jedoch nicht bei den PKW, was darauf hindeutet, dass besonders kurzfristige oder externe Effekte nicht ausreichend erfasst werden. Weihnachten wirkt sich besonders auf den PKW Verkehr aus. Der Verkehr nimmt gegen Ende des Jahres stark ab, während das Modell ein deutlich höheres Verkehrsaufkommen prognostiziert. Das Modell müsste dafür aber die Trendwende um Neujahr herum kennen, wofür der Datensatz von einem Jahr aber nicht ausreicht. Die strukturelle Gleichheit die Voraussetzung für die Anwendung der ARIMA Modelle ist, ist hier nicht gegeben, daher können die Ergebnisse für PKW nur bedingt verlässlich betrachtet werden, während die Modelle für LKW und Busse gute Prognosen liefern.

Die folgende Tabelle fasst die wichtigsten Informationen der besten ARIMA-Modelle je Fahrzeugklasse zusammen und gibt den Skill-Scores im Vergleich zur jeweiligen Benchmark-Methode an.



```

1 library(tibble)
2 # ARIMA-Modelle
3 arima_model <- c(
4   "ARIMA(0,1,1)(2,0,0)[7]",
5   "ARIMA(1,0,1)(3,1,0)[7] manuell",
6   "ARIMA(1,0,1)(2,1,2)[7]"
7 )
8
9 rmse_bench <- c(rmse_pkw_R2_bench, rmse_lkw_R2_bench, rmse_bus_R2_bench)
10 rmse_arima <- c(rmse_pkw_R2_arima, rmse_lkw_R2_arima_man, rmse_bus_R2_arima)
11 skill_score <- round((rmse_bench - rmse_arima) / rmse_bench, 3)
12
13 # Tabelle erstellen
14 arima_summary <- tibble(
15   Fahrzeugklasse = fahrzeugklasse,
16   ARIMA_Modell = arima_model,
17   Benchmark = benchmark,
18   RMSE_Benchmark = round(rmse_bench, 2),
19   RMSE_ARIMA = round(rmse_arima, 2),
20   Skill_Score = skill_score
21 )
22
23 arima_summary

```

```

## # A tibble: 3 x 6
##   Fahrzeugklasse ARIMA_Modell   Benchmark RMSE_Benchmark RMSE_ARIMA Skill_Score
##   <chr>          <chr>          <chr>          <dbl>          <dbl>          <dbl>
## 1 PKW           ARIMA(0,1,1)(2~ SNaive           1170.          1326.          -0.134
## 2 LKW           ARIMA(1,0,1)(3~ Mean              781.           590.           0.245
## 3 Bus           ARIMA(1,0,1)(2~ Mean              43.8           33.8           0.226

```

## 6 Modelle im Vergleich

Insgesamt wurden drei verschiedene Methoden (Naiv, ETS, ARIMA) auf drei Fahrzeugklassen getestet. Das ETS Modell war für die PKW leicht schlechter, während es für die LKW und Busse eine deutliche Verbesserung mit sich brachte, allerdings wurde die Wochensaison nicht vollständig erfasst und Feiertage ebenfalls nicht. Das ARIMA Modell bzw. SARIMA Modell hat sie siebentägige Wochensaison erfasst und bei LKW und Bussen eine noch bessere Prognosegüte geliefert als die ETS Modelle. Das ARIMA Modell hat bei den PKW deutlich schlechter abgeschnitten als bei der naiven Methode, was auf die komplexeren Strukturen und die Weihnachtssaison, die vom Modell in einem Jahr nicht gelernt werden konnte zurückzuführen ist. Für alle Modelle wurden RMSE, MAE, Skill Score etc. verglichen und die Residuen geprüft, die abgesehen von den Feiertagen nicht weiter auffällig waren. Insgesamt haben die ARIMA Modelle am besten performt auch wenn sie keine verbesserte Prognosegüte für die PKW liefern konnten. Für eine detaillierte Übersicht noch einmal die wichtigsten Informationen des besten Modells je Fahrzeugklasse in der folgenden Tabelle:

```

1 skill_score_results = c(0, skill_rmse_lkw_R2_arima_man, skill_rmse_bus_R2_arima) #pkw naiv
2

```

```
3 results <- tibble(  
4   Fahrzeugklasse = fahrzeugklasse,  
5   Benchmark = benchmark,  
6   Modell = c("Snaive", "ARIMA(1,0,1)(3,1,0)[7]", "ARIMA(1,0,1)(2,1,2)[7]"),  
7   Skill_Score = round(skill_score_results, 3)  
8 )  
9  
10 results
```

```
## # A tibble: 3 x 4  
##   Fahrzeugklasse Benchmark Modell          Skill_Score  
##   <chr>          <chr>    <chr>          <dbl>  
## 1 PKW          SNaive   Snaive          0  
## 2 LKW          Mean     ARIMA(1,0,1)(3,1,0)[7] 0.245  
## 3 Bus          Mean     ARIMA(1,0,1)(2,1,2)[7] 0.226
```

## 7 Future Reserch

Worauf bei weiteren Projekten geachtet werden sollte: - mehr Daten über längeren Zeitraum. Auch wenn die Wochenmuster gut erfasst wurden, konnte die Jahressaison nicht erlernt werden, ebenso Feiertage wie Weihnachten, Tag der deutschen Einheit etc. Außerdem können so lokale Trends erlernt werden also dass z. B gegeben Jahresende weniger PKw unterwegs sind und das am Anfang des Jahres wieder zunimmt. - exogene Variablen berücksichtigen. Feiertage haben auch in den ARIMA Modellen zu großen Abweichungen geführt. Feiertage wie Weihnachten können über mehrere Jahre hinweg gelernt werden, Feiertage wie Ostern hingegeben müssten als Dummy Variable mit eingebaut werden bzw. Feiertage generell, da sie den Verkehr (egal welche Klasse) stark beeinflussen. Auch Baustellen könnten als Dummy Variable zu Verbesserungen führen. - Langfristig lohnt sich eine Differenzierung der Fahrzeugklassen, da sich Muster und Sensivitäten unterscheiden. - Erwartungen an Modelle. Die ARIMA Modelle bieten eine solide Basis für Prognosen, stoßen jedoch bei kurzfristigen Schwankungen an ihre Grenzen. Für bessere Prognosen könnten ARIMAX Modelle verwendet werden.

```
1 # Language Options  
2 german: true  
3 lang: de-de
```

## Technical Appendix

```
1 Sys.time()

## [1] "2025-09-03 20:51:08 CEST"

1 sessionInfo()

## R version 4.4.3 (2025-02-28 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=German_Germany.utf8  LC_CTYPE=German_Germany.utf8
## [3] LC_MONETARY=German_Germany.utf8 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.utf8
##
## time zone: Europe/Berlin
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] tibble_3.2.1      feasts_0.4.1      forecast_8.23.0    tidyr_1.3.1
## [5] fable_0.4.1       fabletools_0.5.0  ggplot2_3.5.1     tsibble_1.1.6
## [9] dplyr_1.1.4       fhswf_0.0.9
##
## loaded via a namespace (and not attached):
## [1] ggdist_3.3.2      utf8_1.2.4        generics_0.1.3
## [4] anytime_0.3.11    lattice_0.22-6     digest_0.6.37
## [7] magrittr_2.0.3    evaluate_1.0.3     grid_4.4.3
## [10] timechange_0.3.0  bookdown_0.43      fastmap_1.2.0
## [13] nnet_7.3-20       purrr_1.0.4        scales_1.3.0
## [16] cli_3.6.4         rlang_1.1.5        ellipsis_0.3.2
## [19] munsell_0.5.1     withr_3.0.2        yaml_2.3.10
## [22] tools_4.4.3       parallel_4.4.3     colorspace_2.1-1
## [25] curl_6.2.2        vctrs_0.6.5        R6_2.6.1
## [28] zoo_1.8-13        lifecycle_1.0.4    lubridate_1.9.4
## [31] tseries_0.10-58   urca_1.3-4         pkgconfig_2.0.3
## [34] progressr_0.15.1  pillar_1.10.1      gtable_0.3.6
## [37] quantmod_0.4.26   glue_1.8.0         Rcpp_1.0.14
## [40] xfun_0.52         lmtest_0.9-40      tidyselect_1.2.1
## [43] rstudioapi_0.17.1 knitr_1.50         farver_2.1.2
## [46] nlme_3.1-167      htmltools_0.5.8.1  labeling_0.4.3
## [49] xts_0.14.1        rmarkdown_2.29     timeDate_4041.110
## [52] fracdiff_1.5-3    compiler_4.4.3     quadprog_1.5-8
## [55] TTR_0.24.4        distributional_0.5.0
```