Big Data Analyse mit Python

Ermittlung der Entwicklung der Fahrradtrends in Berlin und Düsseldorf

Das Thema unserer Projektarbeit im Kurs „Big Data Analytics“ ist die Ermittlung der Entwicklung der Fahrradtrends in Berlin und Düsseldorf. Unser Team besteht aus **Nael Al Ahmad, Victoria Blatkowska, Ipek Karakaya** und **Norbert Noupa**.

Die Datensätze von „open.nrw“ und „berlin.de“ enthalten die Dauerzählstellen für den Radverkehr in Düsseldorf und Berlin *(https://open.nrw/dataset/jahresuebersicht-der-dauerzaehlstellen-radverkehr-seit-2012-d, https://www.berlin.de/sen/uvk/verkehr/verkehrsplanung/radverkehr/weitere-radinfrastruktur/zaehlstellen-und-fahrradbarometer/#dauer)*. Für eine Analyse der langfristigen Entwicklung des Radverkehrs über mehrere Jahre in beiden Städten, arbeiten wir mit den Zahlen von 2012-2020.

Teil 1: Bezug herstellen zwischen Daten und Zielerwartung definieren

Wir erwarten ein Zuwachs des Fahrradtrends in den letzten Jahren in Deutschland. Laut der Homepage des „Bundesministerium für Verkehr und digitale Infrastruktur“ *(https://www.bmvi.de/SharedDocs/DE/Artikel/StV/fahrrad-uebersicht.html?https=1)* liegt Fahrrad fahren mehr denn je im Trend. Es entlastet die Umwelt und fördert die eigene sportliche Aktivität, dessen sind sich Menschen immer bewusster. Es wird darauf aufmerksam gemacht, dass rund 80% aller Haushalte in Deutschland mindestens ein Fahrrad besitzen. Es sind etwa 78 Millionen Fahrräder in Deutschland im Einsatz.

Des Weiteren erwarten wir, aufgrund der Pandemie und den damit gegebenen Umständen, einen stärkeren Zuwachs der Radverkehrsdichte von 2019 auf 2020 zu sehen.

Auch die Durchschnittstemperaturen und die Niederschlagsmenge in den unterschiedlichen Jahreszeiten beeinflusst vermutlich, wie viele Leute mit dem Rad unterwegs sind.

Repräsentativ für den Fahrradtrend in Deutschland nehmen wir die Datenmengen von Düsseldorf und Berlin (2012-2020) und überprüfen unsere Vermutungen.

Teil 2: Daten einlesen, bereinigen und abspeichern

Für **Düsseldorf** lesen wir als erstes für jedes Jahr von 2013-2020 die jeweilige csv-Datei ein. Dann bereinigen wir die Daten:

- Duplikate von Messwerten wurden entfernt
- Fahrradfahrten aufsummieren um die Anzahl in Tagen darzustellen
- leere Zellen werden mit dem Wert Null ersetzt (NaN>0)
- ab 2014 gibt es zusätzliche Dauerzählstellen (16Stück)
- im Jahr 2020 kommen zwei weitere Zählstellen dazu, die wir für besseren Vergleich ausgeschnitten haben
- Umlaute in Spaltennamen entfernen für Vereinheitlichung
- manipulierte Datensätze von 1.1.2013 bis 31.12.2020 werden in einer Persistenz-Schicht abgespeichert in ein eigenes DataFrame (als csv)

Für **Berlin** lesen wir als erstes für jedes Jahr von 2017-2020 die jeweilige xlsx-Datei ein. Dann bereinigen wir die Daten:

- Fahrradfahrten aufsummieren um die Anzahl in Tagen darzustellen
- leere Zellen werden mit dem Wert Null ersetzt (NaN>0)
- Umlaute in Spaltennamen entfernen für Vereinheitlichung
- manipulierte Datensätze von 1.1.2017 bis 31.12.2020 werden in einer Persistenz-Schicht abgespeichert in ein eigenes DataFrame (als csv)
- Filter auf farbige Zellen, nicht plausible Daten werden dedektiert in Python siehe Abbildungen unten

Detection of highlighted cells in Excel sheet

	A	B	C	D
1	spalte1	spalte2	spalte3	
2	A2		2,5	
3	A3_rot		3	
4	A4_orange		12,5	
5	A5			
6			17	
7				
8				

```
In [61]: highlighted_cells_df
Out[61]:
```

	spalte1	spalte2	spalte3
0	A2	NaN	2.5
1	A3_rot	NaN	3.0
2	A4_orange	NaN	12.5
3	A5	NaN	NaN
4	NaN	NaN	17.0

```
In [66]: # get sheet names:
         sheet_names = highlighted_cells_wb.sheetnames
         sheet_names
Out[66]: ['Tabelle1']
```

```
In [162]: highlighted_cells_df_clean
Out[162]:
```

	spalte1	spalte2	spalte3
0	A2	NaN	2.5
1	<NA>	NaN	<NA>
2	<NA>	NaN	12.5
3	A5	NaN	<NA>
4	<NA>	NaN	<NA>

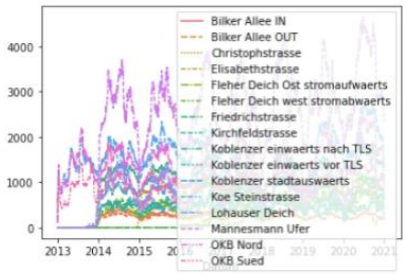
Teil 3: Daten auswerten und visualisieren

Nun sind die Daten vorbereitet, und wir können sie für Düsseldorf anzeigen…

Verlauf der einzelnen Zählstellen

```
In [31]: sns.lineplot(data = bikes_duesseldorf_df_2013_2020_days_roll)
```

```
Out[31]: <AxesSubplot:xlabel='Datum'>
```

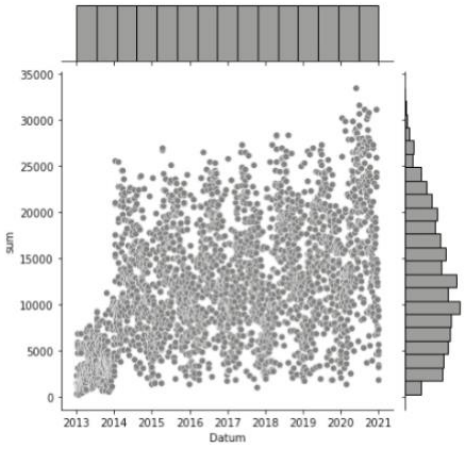


Die tägliche Summe der Werte für alle Zählstationen

```
In [32]: # Zeilenweise addieren:

bikes_duesseldorf_df_2013_2020_days_sum = bikes_duesseldorf_df_2013_2020_days.copy()
bikes_duesseldorf_df_2013_2020_days_sum['sum'] = bikes_duesseldorf_df_2013_2020_days_sum.loc[:].sum(axis=1)
```

```
In [35]: sns.jointplot(x=bikes_duesseldorf_df_2013_2020_days_sum.index,
                       y='sum', data = bikes_duesseldorf_df_2013_2020_days_sum, color = 'grey', kind='scatter')
```

```
Out[35]: <seaborn.axisgrid.JointGrid at 0x127b114f0>
```

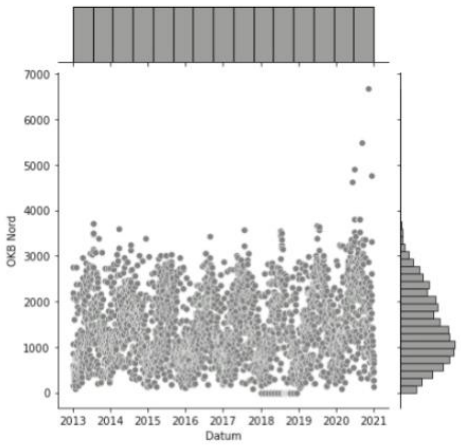


Datenvisualisierung

Darstellung der Werte der letzten 8 Jahre von der zuerst instalierten Zahlstation OKB Nord

In [26]:
```
sns.jointplot(x=bikes_duesseldorf_df_2013_2020_days.index,
              y='OKB Nord', data = bikes_duesseldorf_df_2013_2020_days, color = 'grey', kind='scatter')
```

Out[26]: <seaborn.axisgrid.JointGrid at 0x125a02400>



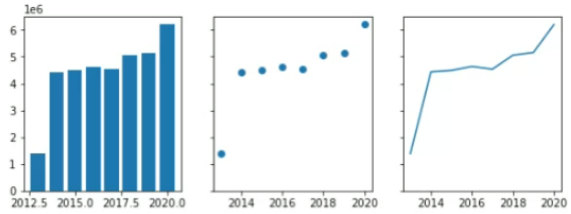
Prozentuale Wachstumsrate der Fahrradfahrer

In [189]:
```
duesseldorf_yearly_data_neu[["sum","Wachstumrate %"]]
```

Out[189]:

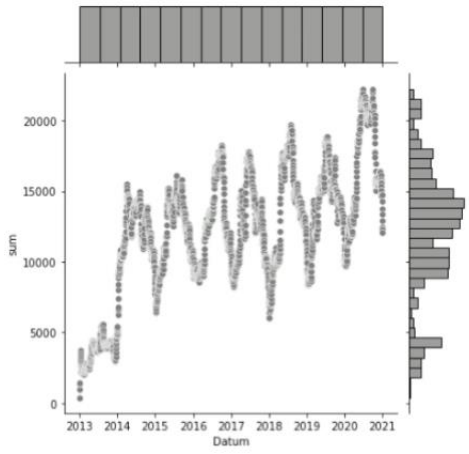
Datum	sum	Wachstumrate %
2013	1383354.0	NaN
2014	4434047.0	220.528729
2015	4488282.0	1.223149
2016	4637735.0	3.329849
2017	4533913.0	-2.238636
2018	5056803.0	11.532864
2019	5155404.0	1.949868
2020	6197238.0	20.208581

```
In [174]: fig, axs = plt.subplots(1, 3, figsize=(9, 3), sharey=True)
          axs[0].bar(duesseldorf_yearly_data_neu.index, duesseldorf_yearly_data_neu["sum"])
          axs[1].scatter(duesseldorf_yearly_data_neu.index, duesseldorf_yearly_data_neu["sum"])
          axs[2].plot(duesseldorf_yearly_data_neu.index, duesseldorf_yearly_data_neu["sum"])

Out[174]: [<matplotlib.lines.Line2D at 0x132a378e0>]
```



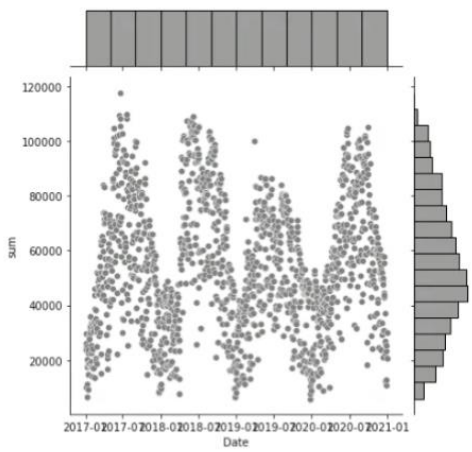

```
In [36]: # gleitender Mittelwert Fenster

         bikes_duesseldorf_df_2013_2020_days_sum_roll = bikes_duesseldorf_df_2013_2020_days_sum.rolling('30D').mean()
```

```
In [37]: sns.jointplot(x=bikes_duesseldorf_df_2013_2020_days_sum_roll.index,
                       y='sum', data = bikes_duesseldorf_df_2013_2020_days_sum_roll, color = 'grey', kind='scatter')

Out[37]: <seaborn.axisgrid.JointGrid at 0x128d7eaf0>
```



Nun sind die Daten vorbereitet, und wir können sie für Berlin anzeigen…

```
In [215]: sns.jointplot(x = 'Date', y = 'sum', data = bikes_berlin_df_total_days, color = 'grey')

Out[215]: <seaborn.axisgrid.JointGrid at 0x136007a30>
```



Teil 4: Zielerwartung (aus Teil1) überprüfen und Ergebnis erläutern

Wir können feststellen, dass es einen relativ stabilen Trend im Jahreszeitenverlauf gibt: An trockenen warmen Tagen sind mehr Fahrradfahrer unterwegs, als an nassen kalten Wintertagen.

Mitte Februar 2020 kam es zum Shutdown in China und anderen Teilen Asiens. Lieferketten waren gestört bzw. unterbrochen. Hersteller von Fahrrädern und Zubehör konnten nur eingeschränkt produzieren. Mitte März 2020 wurden dann die Fahrradgeschäfte in den meisten Bundesländern in Deutschland geschlossen. Der Verkauf wurde stark eingeschränkt. Nach der Öffnung der Geschäfte jedoch, boomt die Nachfrage nach Produkten der deutschen und internationalen Fahrradindustrie. Laut dem „Zweirad-Industrie-Verband“ sind einige Gründe dafür: Das Vermeiden der Öffentlichen Verkehrsmittel, der Drang nach Bewegung an der frischen Luft mit ausreichendem Abstand und geänderte Urlaubspläne. Die Leute sind sich der Vorteile des Radfahrens bewusst: Es ist infektionssicher und stärkt das Immunsystem.

Der Leiter im Marketing & Kommunikation des Zweirad-Industrie-Verbandes David Eisenberger sagt: „Fahrradmobilität ist systemrelevant. Das haben die letzten Monate gezeigt. Darüber hinaus wissen wir, dass Fahrrad und E-Bike in diesem Jahr zusätzlich neue Zielgruppen angesprochen haben. Viele dieser neuen Nutzer werden die Zweiradmobilität auch nach der Krise nicht mehr missen wollen. Wir sind vorsichtig optimistisch, dass die Branche auch im Jahr 2021 auf einem hohen Niveau, ähnlich des Vorjahres, abschließen kann.“ *(https://www.ziv-zweirad.de/presse-medien/pressemitteilungen/detail/?tx_ttnews%5Btt_news%5D=950&cHash=9265b5aa4b39e10bac9f9912ce8bde67)*

Allerdings zeigen diese Zahlen nur einen Trend: Denn insbesondere Baustellen können zu Unschärfen in der Statistik und somit Ausreißern nach oben und unten führen. Sofern in der Darstellung Angaben fehlen, war die entsprechende Dauerzählstelle zum Messzeitpunkt außer Betrieb.

So kann es sein, dass am Tag X wenige Radfahrer gezählt werden, am gleichen Tag ein Jahr später mehr gezählt werden, da dann evtl. das Wetter besser ist oder es sich um einen Wochenendetag handelt.

```
In [230]: berlin_yearly_data_neu[["sum","Wachstumrate %"]]
```

Out[230]:

Date	sum	Wachstumrate %
2017	19631392.0	NaN
2018	21369545.0	8.853947
2019	18024377.0	-15.653904
2020	20485266.0	13.653115

```
In [232]: duesseldorf_yearly_data_neu[["sum","Wachstumrate %"]]
```

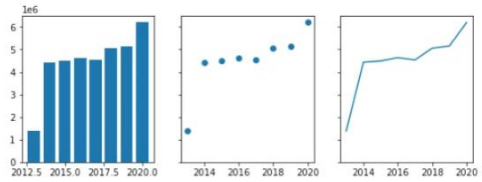
Out[232]:

Datum	sum	Wachstumrate %
2013	1383354.0	NaN
2014	4434047.0	220.528729
2015	4488282.0	1.223149
2016	4637735.0	3.329849
2017	4533913.0	-2.238636
2018	5056803.0	11.532864
2019	5155404.0	1.949868
2020	6197238.0	20.208581

Vergleich Düsseldorf - Berlin

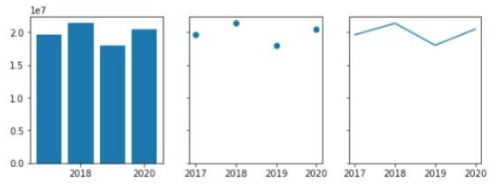
```
In [231]: fig, axs = plt.subplots(1, 3, figsize=(9, 3), sharey=True)
axs[0].bar(duesseldorf_yearly_data_neu.index, duesseldorf_yearly_data_neu["sum"])
axs[1].scatter(duesseldorf_yearly_data_neu.index, duesseldorf_yearly_data_neu["sum"])
axs[2].plot(duesseldorf_yearly_data_neu.index, duesseldorf_yearly_data_neu["sum"])
```

```
Out[231]: [<matplotlib.lines.Line2D at 0x12ad38d60>]
```




```
In [225]: fig, axs = plt.subplots(1, 3, figsize=(9, 3), sharey=True)
axs[0].bar(berlin_yearly_data_neu.index, berlin_yearly_data_neu["sum"])
axs[1].scatter(berlin_yearly_data_neu.index, berlin_yearly_data_neu["sum"])
axs[2].plot(berlin_yearly_data_neu.index, berlin_yearly_data_neu["sum"])
```

```
Out[225]: [<matplotlib.lines.Line2D at 0x125c56b50>]
```



Visualisierung der Coronafallzahlen in Berlin

```
In [200]: sns.jointplot(x = 'Datum', y = 'Fallzahl', data = corona_berlin_df, color = 'grey')
```

```
Out[200]: <seaborn.axisgrid.JointGrid at 0x130d47460>
```