ToiletPaper #108

ERP-Kung-Fu mit Pandas

Autoren: Dominik Jain & Kristof Schröder / Senior Software Architect & Software Architect / Business Division New Business

Wir können uns die Möglichkeiten moderner Analytics-Bibliotheken zunutze machen, um detailliertere Einsichten über Projekte zu erlangen als uns das ERP dies von Haus aus erlauben würde. Insbesondere die Python-Bibliothek pandas erfreut sich derzeit großer Beliebtheit und findet im Analytics-Bereich häufig Anwendung.
Sie erlaubt es unmittelbar, Excel-Exports des ERPs zu laden, geeignet zu transformieren und zu visualisieren. Einen nutzbaren ExcelExport aus dem ERP erhält man, indem man in der Projektverwaltung unter Verwalten/Gebuchte Posten die Option zum Öffnen in Office verwendet. Es wird die aktuell sichtbare Ansicht exportiert. Je nach Anwendungsfall kann es notwendig sein, weitere Felder (wie die Aktivitätsnummer oder die Stundenmenge) vor dem Export einzublenden.

✓ Unleash the Pandas

Wir laden den ERP-Export als *pandas.DataFrame*:

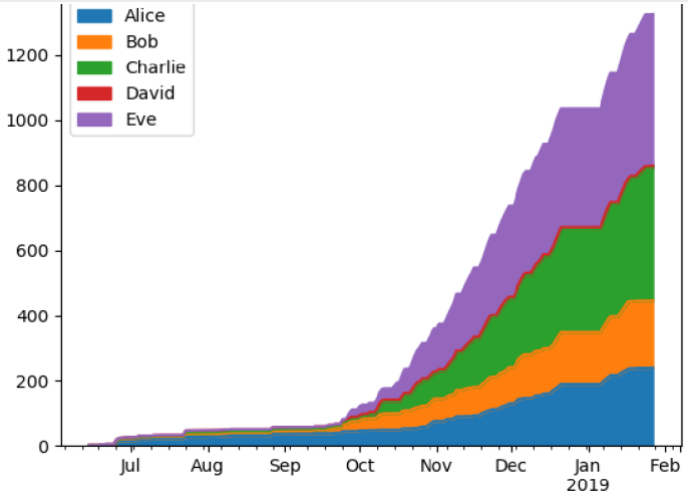
```
import pandas as pd
df = pd.read_excel("Projektbuchungen_636843550793374935.xlsx")
```

Wir interessieren uns für den Verlauf der gebuchten Stunden, aufgespaltet nach verschiedenen Attributen. Das Ziel ist, einen neuen DataFrame zu erstellen, der nur die darzustellenden Zeitreihen enthält. Dieser soll einen Zeitindex verwenden, der den betrachteten Zeitraum umfasst. Wir spalten die Daten über die groupby-Operation nach dem Wert einer Spalte auf, gruppieren die Stundenmenge nach Datum, summieren die Einträge je eines Tages und reindizieren, so dass sich jede neue Zeitreihe über den gleichen Zeitraum erstreckt. Dann fassen wir alle Zeitreihen zu einem DataFrame zusammen und plotten diesen:

```
def plot_hours_by_column(df, col, cumulative, plot_kind):
    dates = df["Datum"]
    full_index = pd.DatetimeIndex(start=min(dates), end=max(dates),
                                  freq='1D')
    series = {}
    for key, pdf in df.groupby(df[col]):
        hours = pdf["Menge"].groupby(pdf["Datum"]).sum() \
                .reindex(index=full_index, fill_value=0)
        hours = hours.cumsum() if cumulative else hours
        series[key] = hours
    pd.DataFrame(series).plot(kind=plot_kind)
```

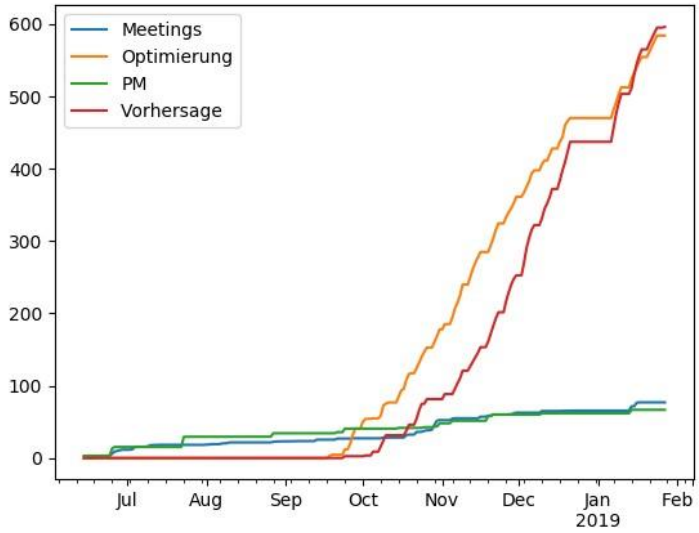
Diese Funktion können wir verwenden, um die von den einzelnen Teammitgliedern („Ressourcen“, Namen verändert) akkumulierten Stunden über die Zeit zu plotten:

```
plot_hours_by_column(df, "Ressourcenname", True, "area")
```



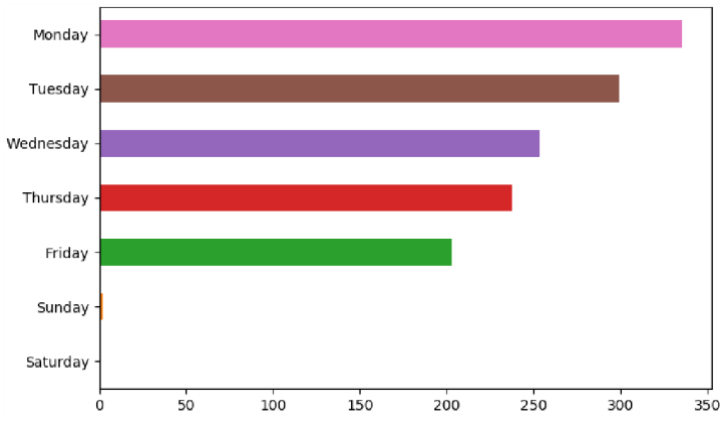
Als nächstes plotten wir den Verlauf der Arbeitszeit, die auf die verschiedenen Aktivitäten entfallen ist. Da das ERP nur die Aktivitätsnummern exportieren kann, bilden wir diese vorher auf die entsprechenden Namen ab.

```
act_names = {"AK0005412": "Vorhersage",
             "AK0005413": "Optimierung", ...}
df["Aktivität"] = [act_names.get(n, n)
                   for n in df["Aktivitätsnummer"]]
plot_hours_by_column(df, "Aktivität", True, "line")
```



Ferner interessieren wir uns für die Verteilung der Arbeitsstunden über die Wochentage:

```
df["Menge"].groupby(df["Datum"].dt.weekday_name).sum() \
    .sort_values().plot(kind='barh')
```



+ Weiterführende Aspekte

Wir haben angefangen, eine Bibliothek zu entwickeln, die nützliche Funktionen geeignet abstrahiert:
https://bitbucket.jambit.com/users/djain/repos/erp-analytics/browse - Pull-Requests willkommen!