

DEFINITION

Das Parkinson-Syndrom (PD) ist eine komplexe, degenerative Erkrankung, die die für motorische Bewegungen zuständigen Nervenzellen angreift [2].

PROBLEM

Die Herausforderung bestand darin, die 5'875 Stimmaufnahmen zu analysieren und miteinander in Beziehung zu setzen.

DATENSATZ

Der CSV-Datensatz umfasst 16 Stimmmessungen von 42 Parkinson-Patienten im Frühstadium. Wir verfügen über insgesamt 5'875 Aufnahmen zur Analyse der Sprachkompetenzen und deren Vergleichbarkeit. Pro Proband wurden zwischen 120 und 180 Aufzeichnungen durchgeführt (siehe Abb. 1). Die durchschnittliche Anzahl der Messungen beträgt 139,8. Hauptziel der Daten ist die Vorhersage der UPDRS-Werte aus diesen Stimmessungen [5]. Die "Unified Parkinson's Disease Rating Scale" (UPDRS) ist eine Punkteskala von 0 (keine Beeinträchtigung) bis 199 (schwerste Beeinträchtigung), die anhand von Fragebogen und klinischer Untersuchung den Parkinson-Verlauf bewertet [6].

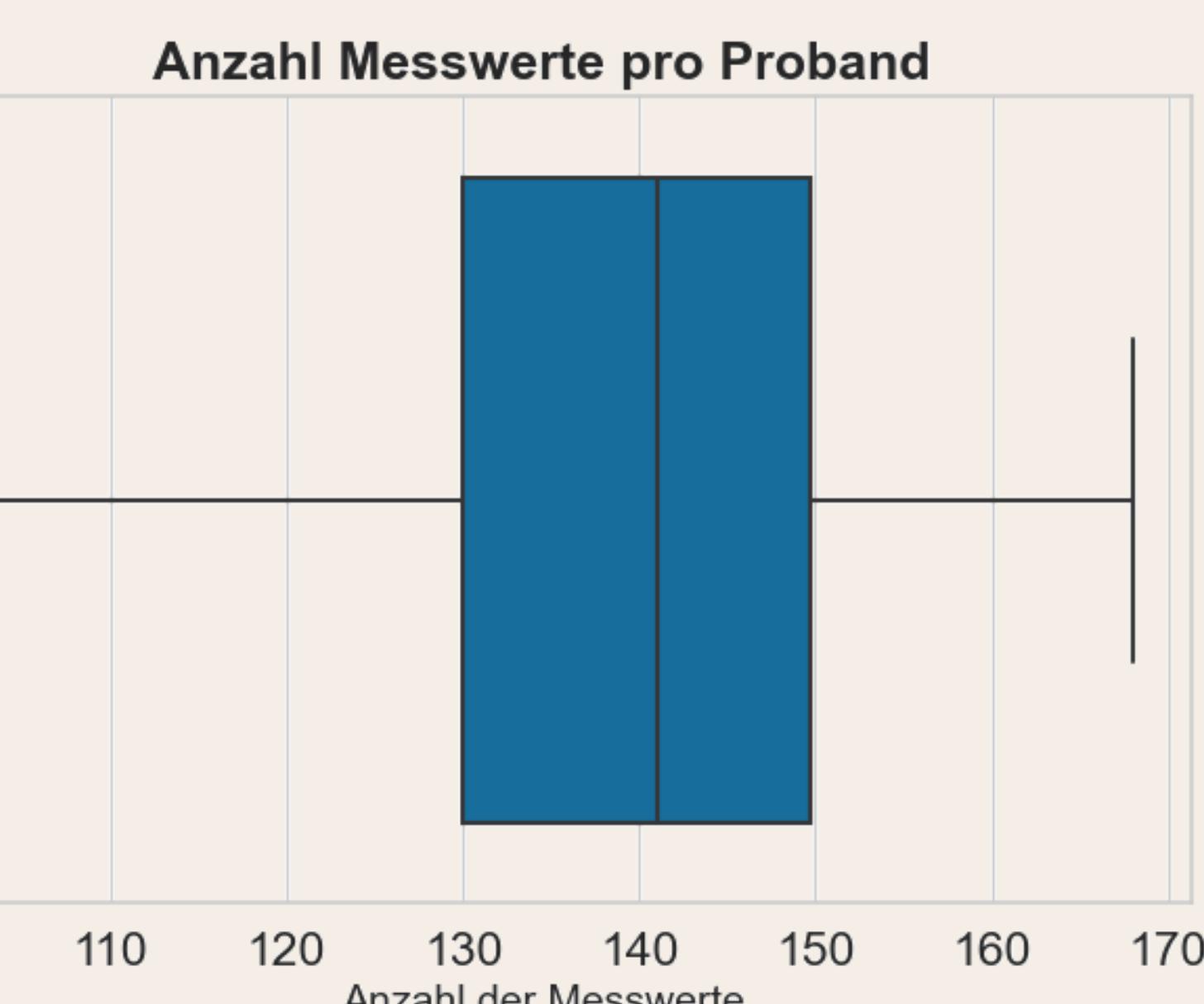


Abb. 1: Boxplot der Anzahl der Messwerte pro Proband. Die Box repräsentiert das Interquartilbereich und der Median ist als Linie innerhalb der Box dargestellt. Die „Whiskers“ zeigen die Spannweite der Daten ausserhalb der Quartile, ohne betrachtete Ausreißer.
Quelle: Eigene Darstellung.

LEGENDE [3]

motor_UPDRS: Motorischer Teil der UPDRS

total_UPDRS: Gesamtpunktzahl der UPDRS

Jitter: Verschiedene Masse für Frequenzschwankungen in der Stimme

Shimmer: Masse für Amplitudenschwankungen in der Stimme

NHR: Noise to Harmonics Ratio

HNR: Harmonics to Noise Ratio

RPDE: Recurrence Period Density Entropy

DFA: Detrended Fluctuation Analysis

PPE: Pitch Period Entropy

TEILNEHMER

Unter den 42 Probanden sind 28 männlich (M) und 14 weiblich (W), was die höhere Betroffenheit von Männern unterstreichen könnte. Laut Abbildung 2 liegt das Medianalter beider Geschlechter bei etwa 60 Jahren. Die breitere Altersverteilung bei Männern könnte auf unterschiedliche Krankheitsmuster hindeuten.

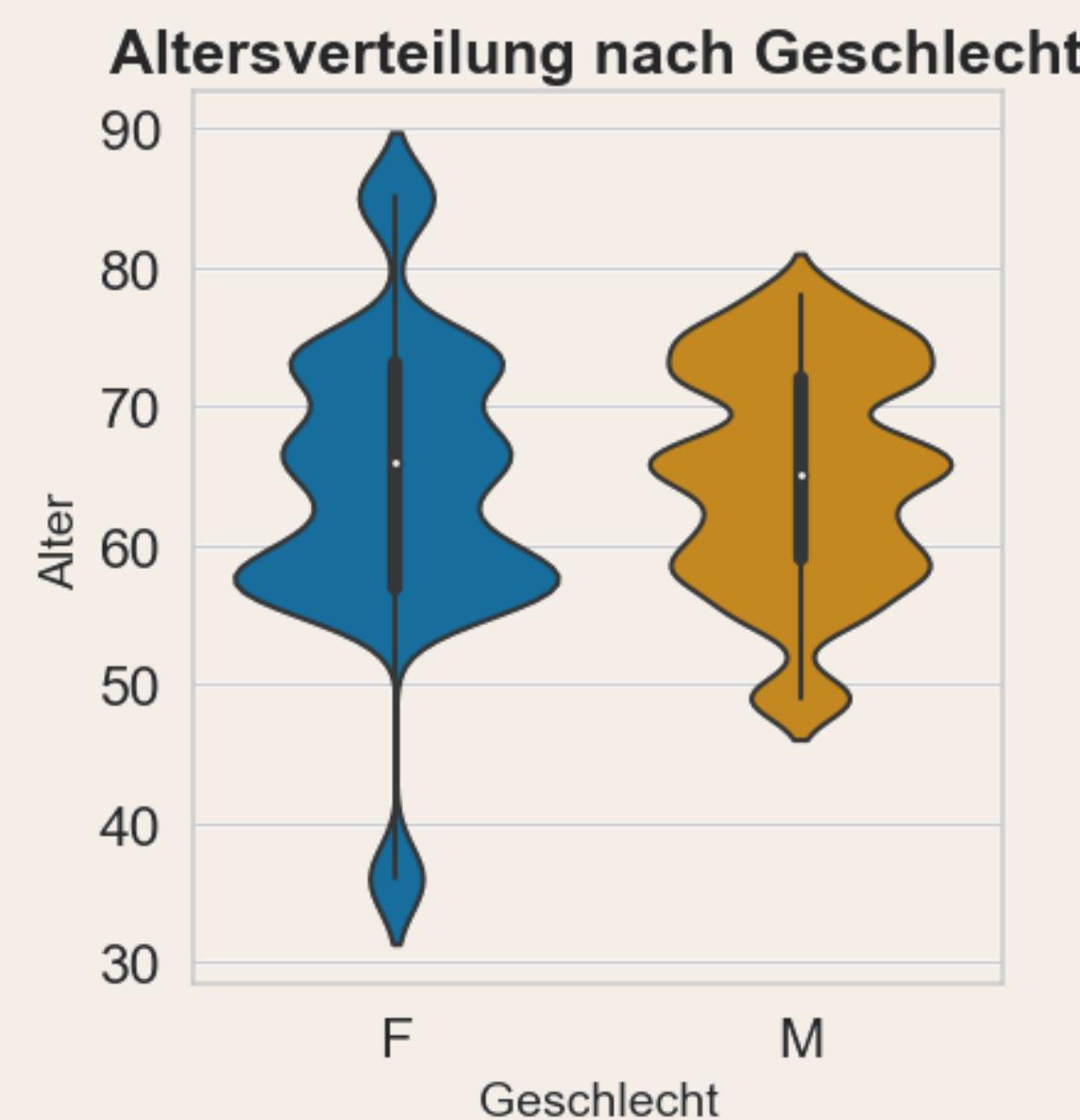


Abb. 2: Altersverteilung nach Geschlecht, dargestellt in Violin-Plots mit einer horizontalen Linie, die den Median des Alters anzeigen.
Quelle: Eigene Darstellung.

LÖSUNG

Interaktionsvariablen [7], gebildet aus motorischen UPDRS-Werten multipliziert mit Jitter und Shimmer [8] und dargestellt in Abbildung 3, liefern tiefere Einblicke in den Krankheitsverlauf von Parkinson. Diese Variablen könnten als aussagekräftigere Indikatoren für die Krankheitsdynamik dienen und in Vorhersagemodellen verwendet werden, um die UPDRS-Werte genauer zu prognostizieren [8]. Für eine zuverlässige Vorhersage müssen diese Variablen jedoch in ein statistisches Modell oder einen maschinellen Lernalgorithmus integriert und validiert werden. Mögliche Modelle umfassen lineare Regression, Lasso-Regression oder Ridge-Regression wie auch nicht-lineare Regressionsmodelle.

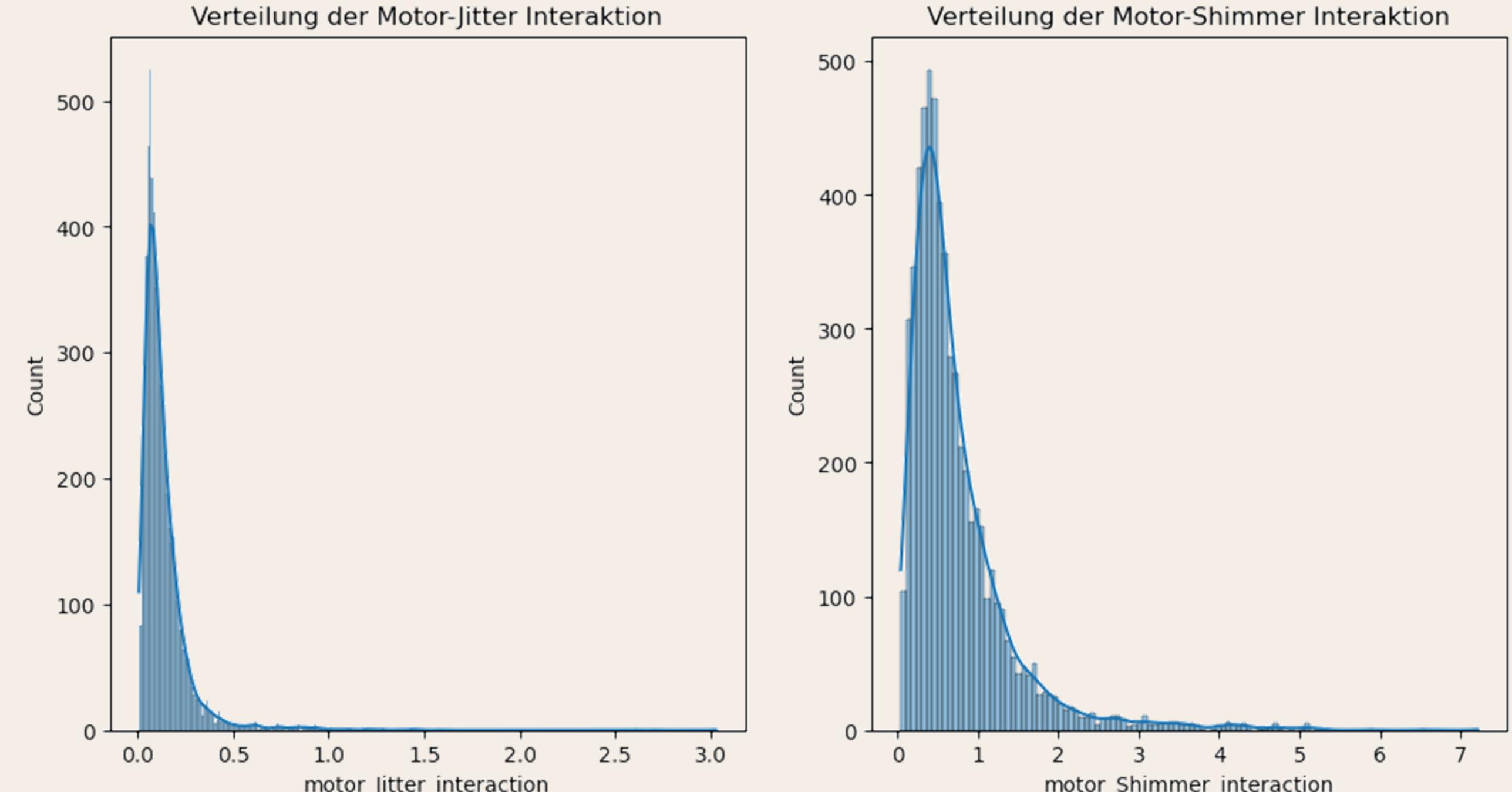


Abb. 3: Histogramme der Verteilung der motorischen 'Jitter' (links) und der 'Shimmer' Interaktion (rechts) bei Stimmessungen. Jede Säule repräsentiert die Anzahl der Messungen in der entsprechenden Kategorie.
Quelle: Eigene Darstellung.

GEWINN

Die Scatterplots in der Abbildung 4 zeigen geschlechtsspezifische Muster in den Stimmparametern von Parkinson-Patienten: Frauen (blaue Punkte) haben generell höhere Werte als Männer (orange Punkte), da Frauen eine höhere Stimmfrequenz aufweisen. Es ist eine positive Korrelation zwischen Jitter (%) und Shimmer zu erkennen, was darauf hindeutet, dass eine grösere Unregelmässigkeit in der Stimmlage mit einer höheren Amplitude-Variabilität verbunden ist. HNR korreliert mit dem Jitter Wert negativ. Das deutet darauf hin, dass eine grösere Unregelmässigkeit in der Stimmlage mit einer geringeren Stimmqualität verbunden ist. Eine hohe HNR, die für mehr Rauschen in der Stimme steht, korreliert negativ mit HNR, dem Indikator für Stimmklarheit, was auf eine geringere Qualität der Stimme hinweist.

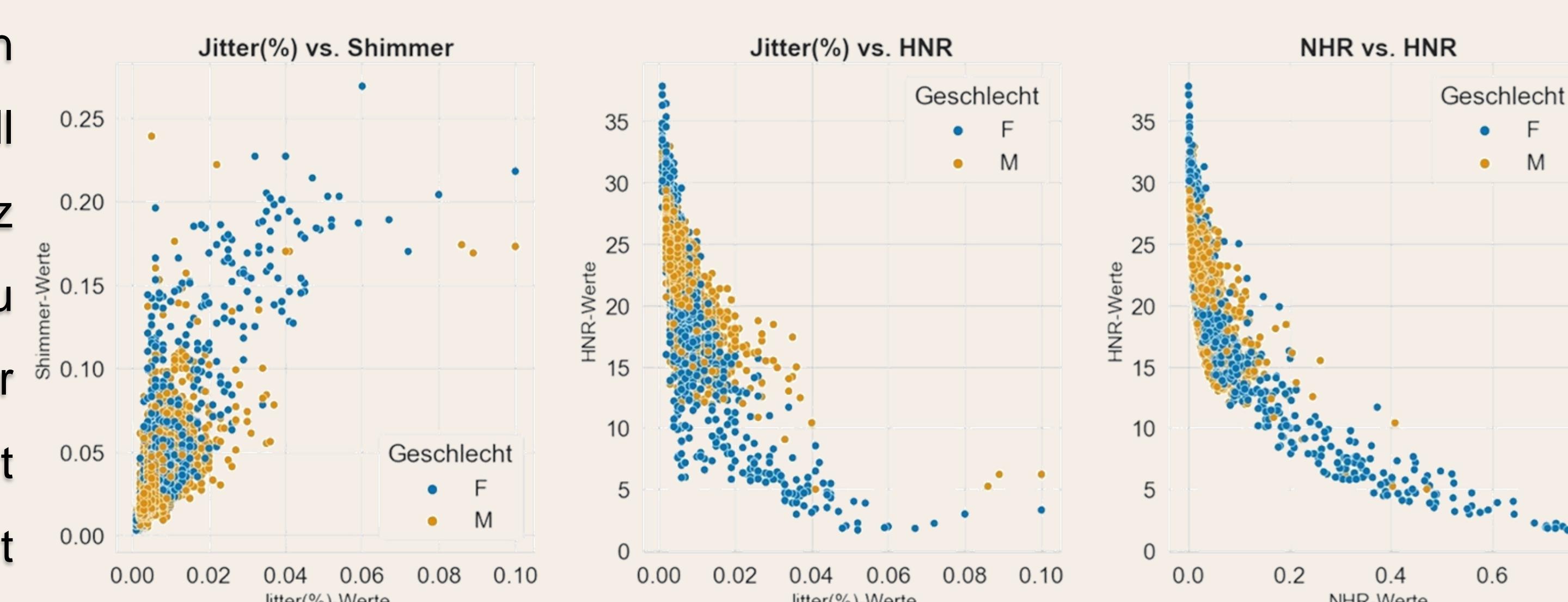
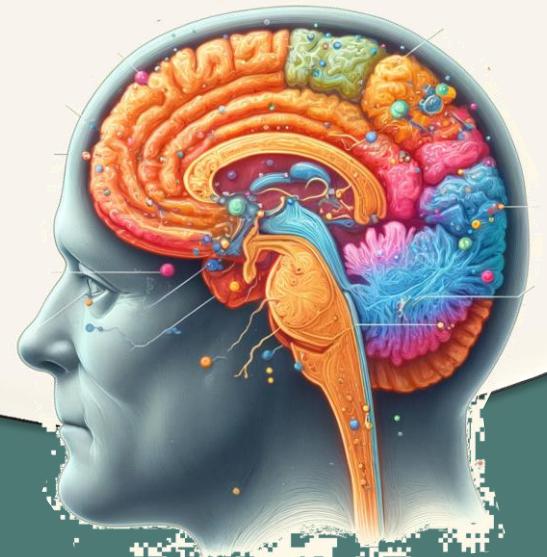


Abb. 4: Scatterplots zum Vergleich der Stimmparameter Jitter (%), Shimmer und Harmonic-to-Noise Ratio (HNR) von Parkinson-Patienten. Die Daten sind nach Geschlecht unterschieden, mit blauen Punkten für Frauen (F) und orangen Punkten für Männer (M).
Quelle: Eigene Darstellung.

QUELLEN

- [1] M. A. Little, P. E. McSharry, S. J. Roberts, D. A. Costello, und I. M. Moroz, „Exploiting Nonlinear Recurrence and Fractal Scaling Properties for Voice Disorder Detection“, Biomed. Eng. OnLine, Bd. 6, Nr. 1, S. 23, 2007, doi: 10.1186/1475-925X-6-23.
- [2] S. Zhao, J. Zhang, und J. Zhang, „Predicting UPDRS in Parkinson's disease using ensembles of self-organizing map and neuro-fuzzy“, J. Cloud Comput., Bd. 13, Nr. 1, S. 83, Apr. 2024, doi: 10.1186/s13677-024-00641-9.
- [3] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman, und L. O. Ramig, „Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's Disease“, IEEE Trans. Biomed. Eng., vol. 56, no. 4, pp. 1015-1022, Apr. 2009, doi: 10.1109/TBME.2008.2005954.
- [4] A. Tsanas, M. A. Little, P. E. McSharry, und L. O. Ramig, "11_parkinsons_updrs.data" [Online]. Verfügbar: https://moodle.fhnw.ch/pluginfile.php/3004530/mod_resource/content/0/11_parkinsons_updrs.data. [Zugriffen: 26.04.2024].
- [5] A. Tsanas, M. A. Little, P. E. McSharry, und L. O. Ramig, "11_parkinsons_updrs.names", 2009. [Online]. Verfügbar: https://moodle.fhnw.ch/pluginfile.php/3004531/mod_resource/content/0/11_parkinsons_updrs.names. [Zugriffen: 26.04.2024].
- [6] Pschyrembel Klinisches Wörterbuch, 264. Auflage. De Gruyter, 2013.
- [7] R. Grünwald, „Interaktionseffekte in Stata einfach erklärt und berechnet - NOVUSTAT“, Novustat, 11. Februar 2021. [Online]. Verfügbar: <https://novustat.com/statistik-blog/interaktionseffekte-in-stata.html> [Zugriffen: 26.04.2024].
- [8] A. Suppa u. a., „Voice in Parkinson's Disease: A Machine learning study“, Frontiers in Neurology, Bd. 13, Feb. 2022, doi: 10.3389/fneur.2022.831428.



Pre-Processing

```
# Laden der Libraries und der CSV Datensatz Datei
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('Parkinson.csv')
df
print(df.info())
print(df.head())

# Kontrollieren, ob die Spalten brauchbare Werte hat und Liste der zu überprüfenden Spalten.
columns_to_check = ['subject#', 'age', 'sex', 'test_time', 'motor_UPDRS', 'total_UPDRS', 'Jitter(%)',
'Jitter(Abs)', 'Jitter:RAP', 'Jitter:PPQ5', 'Jitter:DDP', 'Shimmer', 'Shimmer(dB)', 'Shimmer:APQ3',
'Shimmer:APQ5', 'Shimmer:APQ11', 'Shimmer:DDA', 'NHR', 'HNR', 'RPDE', 'DFA', 'PPE']

# Funktion zur Überprüfung einer Spalte auf negative Werte und NaN-Werte
def check_column_plausibility(column_name):
    negative_values_count = (df[column_name] < 0).sum()
    nan_values_count = pd.isna(df[column_name]).sum()
    summary = df[column_name].describe()
    return negative_values_count, nan_values_count, summary

# Durchführen der Überprüfungen
for column in columns_to_check:
    negative_values_count, nan_count, summary = check_column_plausibility(column)
    print(f"Überprüfung für '{column}':")
    print(f"Anzahl negativer Werte: {negative_values_count}")
    print(f"Anzahl von NaN Werten: {nan_count}")
    print(f"Statistische Zusammenfassung:\n{summary}\n")

# Datentypumwandlung der Spalte 'subject#' von int64 zu 'category', da diese Spalte zur eindeutigen
# Kennzeichnung der Teilnehmer dient und nicht für mathematische Operationen verwendet wird.
df['subject#'] = df['subject#'].astype('category')

# Anzahl Teilnehmer*innen:
unique_participants = df['subject#'].nunique()
print(f"Anzahl der eindeutigen Teilnehmer: {unique_participants}")

# Wie viele Zeilen (Messpunkte) jeder 'subject#' (Testperson) zugeordnet sind, absteigend nach Anzahl
# Messpunkte
observations_per_subject = df['subject#'].value_counts()
print("Anzahl der Messpunkte pro Testperson:")
print(observations_per_subject)

# 'sex' Datentypumwandlung zu "category", da wie bei 'subject#' nicht für mathematische Operationen gedacht
# ist.
df['sex'] = df['sex'].astype('category')

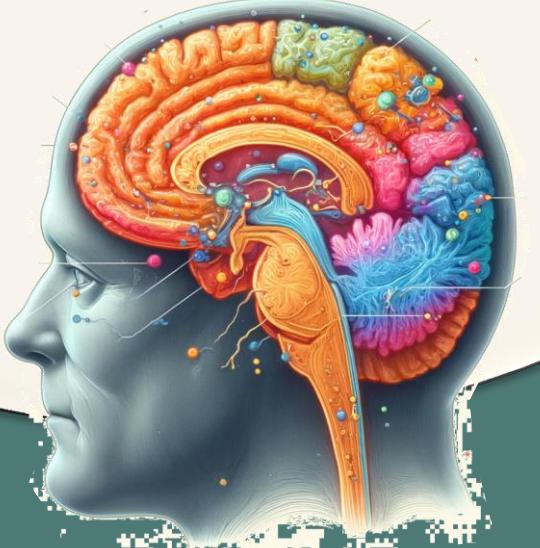
# Kontrollieren ob Spalte "sex" 0 und 1 Werte hat oder andere die nicht korrekt sind → "0" = männlich, "1" =
# weiblich
df.sex.value_counts()

# 0 und 1 nach M und F ersetzen, da es sonst einen Mittelwert gibt, den wir in der Statistik nicht wollen.
df['sex'] = df['sex'].replace({0: 'M', 1: 'F'})

# Überprüfung der Datentypumwandlung
print(df['sex'].value_counts())

# Einzigartige Werte und deren Anzahl in 'sex'
unique_sex_values = df['sex'].unique()
print(f"Einzigartige Werte in 'sex': {unique_sex_values}")

# Durchschnittswerte jeder Spalte pro Teilnehmer
for subject_num in df['subject#'].unique():
    df_subject = df[df['subject#'] == subject_num]
    numerical_columns = df_subject.select_dtypes(include=[np.number]).columns.tolist()
    mean_values = df_subject[numerical_columns].mean()
    print("Mean values for Subject", subject_num, ":")
    print(mean_values)
    print("\n")
```



Pre-Processing

```
# Bei der Durchführung der Überprüfungen haben wir festgestellt, dass 'test_time' negative Werte enthält, die auf NaN gesetzt werden sollten und überprüfen, ob noch weitere NaN Werte vorhanden sind.
print(df.isnull().sum())

# Nochmals bei allen Spalten spezifisch nach negativen und zusätzlich 0 Werte suchen.
numerical_df = df.select_dtypes(include=[np.number])
zero_counts = (numerical_df <= 0).sum()
print("Anzahl der negativen- und 0-Werten in jeder numerischen Spalte:")
print(zero_counts)

# Filtern der Werte, bei denen 'test_time' zwischen 0 und 1 liegt, da weniger als eine Sekunde nicht realistisch ist.
filtered_values = df[(df['test_time'] >= 0) & (df['test_time'] <= 1)]
print(filtered_values['test_time']) # sind keine Werte zwischen 0 und 1 vorhanden

# Negative und 0 Werte ausgeben lassen von test_time
negative_test_time = df[df['test_time'] <= 0]
if negative_test_time.empty:
    print("Keine falsche Testzeit gefunden.")
else:
    print(f"Falsche Testzeit:\n{negative_test_time['test_time']}")

# Ersetze negative Werte in 'test_time' mit NaN
df.loc[df['test_time'] <= 0, 'test_time'] = np.nan
print("Aktualisierte 'test_time' Spalte:")
print(df['test_time'])

# NaN Werte von test_time ausgeben
nan_values_test_time = df[df['test_time'].isnull()]['test_time']
print(nan_values_test_time)

# Nochmals kontrollieren, ob im Datensatz keine weiteren negative Werte vorhanden sind
numeric_df = df.select_dtypes(include=[np.number])
has_negative_values = (numeric_df <= 0).any().any()
print(has_negative_values)

# Datensatz auf Duplikate prüfen
duplicates = df.duplicated()
print(f"Im Datensatz sind {str(duplicates.sum())} Duplikate vorhanden") # keine vorhanden

# Interaktionsmerkmal aufzeigen, wie sich motor_UPDRS und Jitter und Shimmer gegenseitig beeinflussen
df['motor_Jitter_interaction'] = df['motor_UPDRS'] * df['Jitter(%)']
df['motor_Shimmer_interaction'] = df['motor_UPDRS'] * df['Shimmer']

# Speichern des cleaned Dataset für Visualisierung
df.to_csv('preprocessed_dataset.csv', index=False)
```

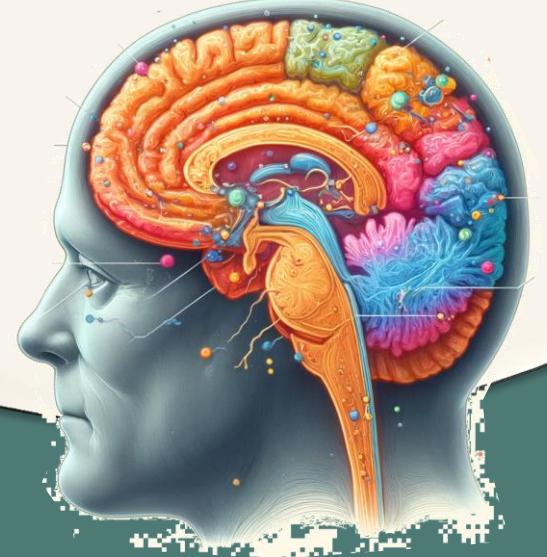
Visualisierung

```
# Laden der Libraries und laden der CSV Datei
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('preprocessed_dataset.csv')
df

# Den visuellen Stil für den Plot festlegen, setzen des Kontexts für die Skalierung der Plot-Elemente
sns.set_style("whitegrid")
sns.set_context("notebook", font_scale=1.5)

# Die farbenblind freundliche Farbpalette für das Diagramm festlegen.
sns.set_palette("colorblind")

# Matplotlib-Stil-Anpassungen
plt.rc('figure', titlesize=18) # Titelgrösse für alle Plots
plt.rc('axes', titlesize=18) # Titelgrösse für Achsen
plt.rc('axes', labelsize=14) # Beschriftungsgrösse für Achsenbeschriftungen
```



Visualisierung

```
# Erstellen des Balkendiagramms 'Anzahl Messungen pro Geschlecht' (siehe Abb. 5)
plt.figure(figsize=(8, 5))
bar_plot = sns.barplot(x=measurement_counts.index, y=measurement_counts.values)
plt.title('Anzahl Messungen pro Geschlecht', fontweight='bold')
plt.xlabel('Geschlecht')
plt.ylabel('Anzahl der Messungen')
```

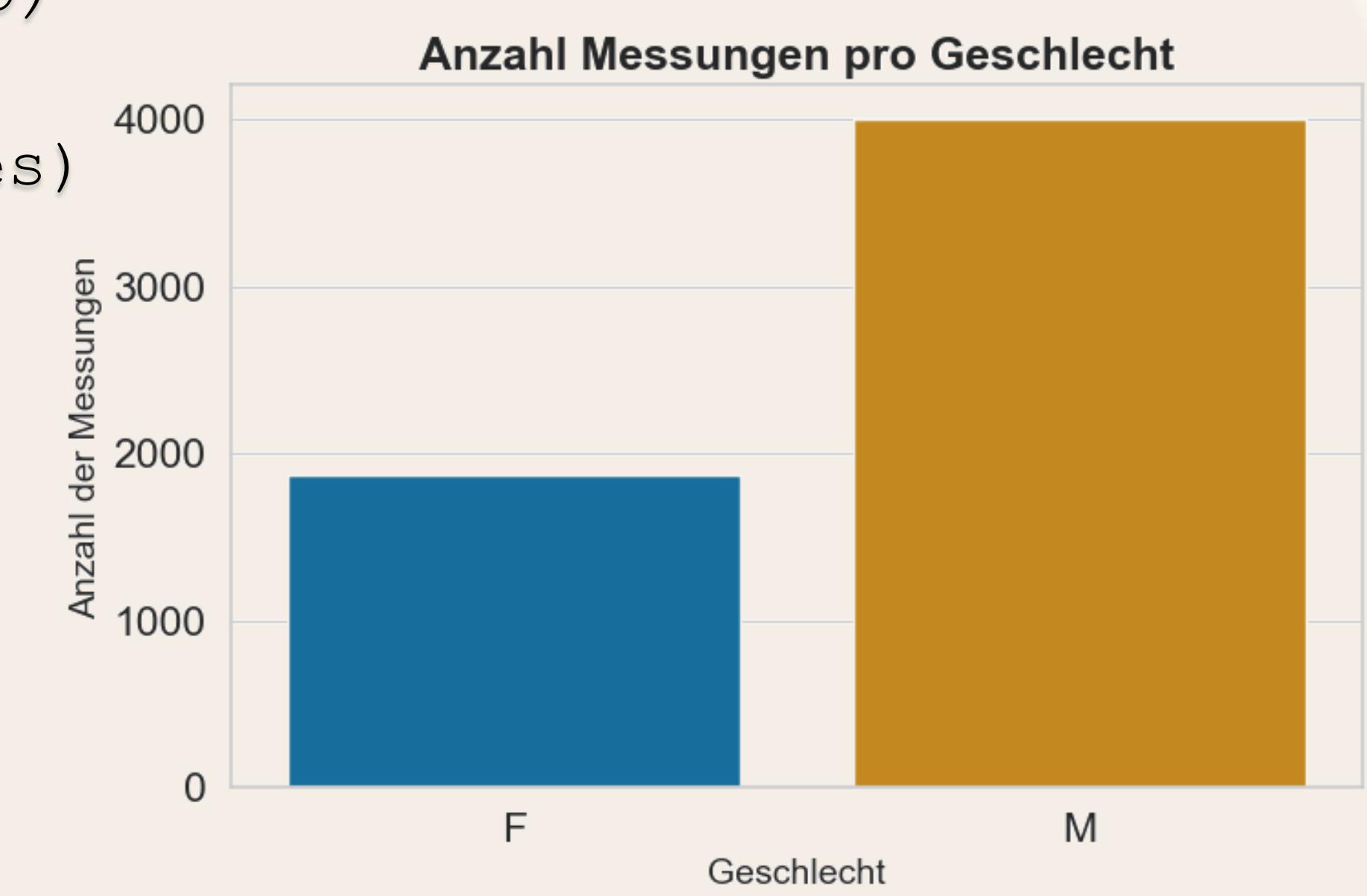


Abb. 5: Anzahl der Messungen pro Geschlecht, mit der Anzahl der durchgeföhrten Messungen für weibliche (F) und männliche (M) Probanden.
Quelle: Eigene Darstellung.

```
# Erstellen des Violin Plot 'Altersverteilung nach Geschlecht'
# (siehe Hauptposter, Abb.2)
plt.figure(figsize=(8, 5))
plt.subplot(1, 2, 2) # 1 row, 2 columns, 2nd subplot
sns.violinplot(x='sex', y='age', data=df)
plt.title('Altersverteilung nach Geschlecht', fontweight="bold")
plt.xlabel('Geschlecht')
plt.ylabel('Alter')
plt.tight_layout()
```

```
# Erstellen des Box Plot 'Anzahl Messwerte pro Proband' (siehe Hauptposter, Abb.1)
plt.figure(figsize=(8, 5))
sns.boxplot(x=measurements_per_subject)
plt.title('Anzahl Messwerte pro Proband', fontweight='bold')
plt.xlabel('Anzahl der Messwerte')
```

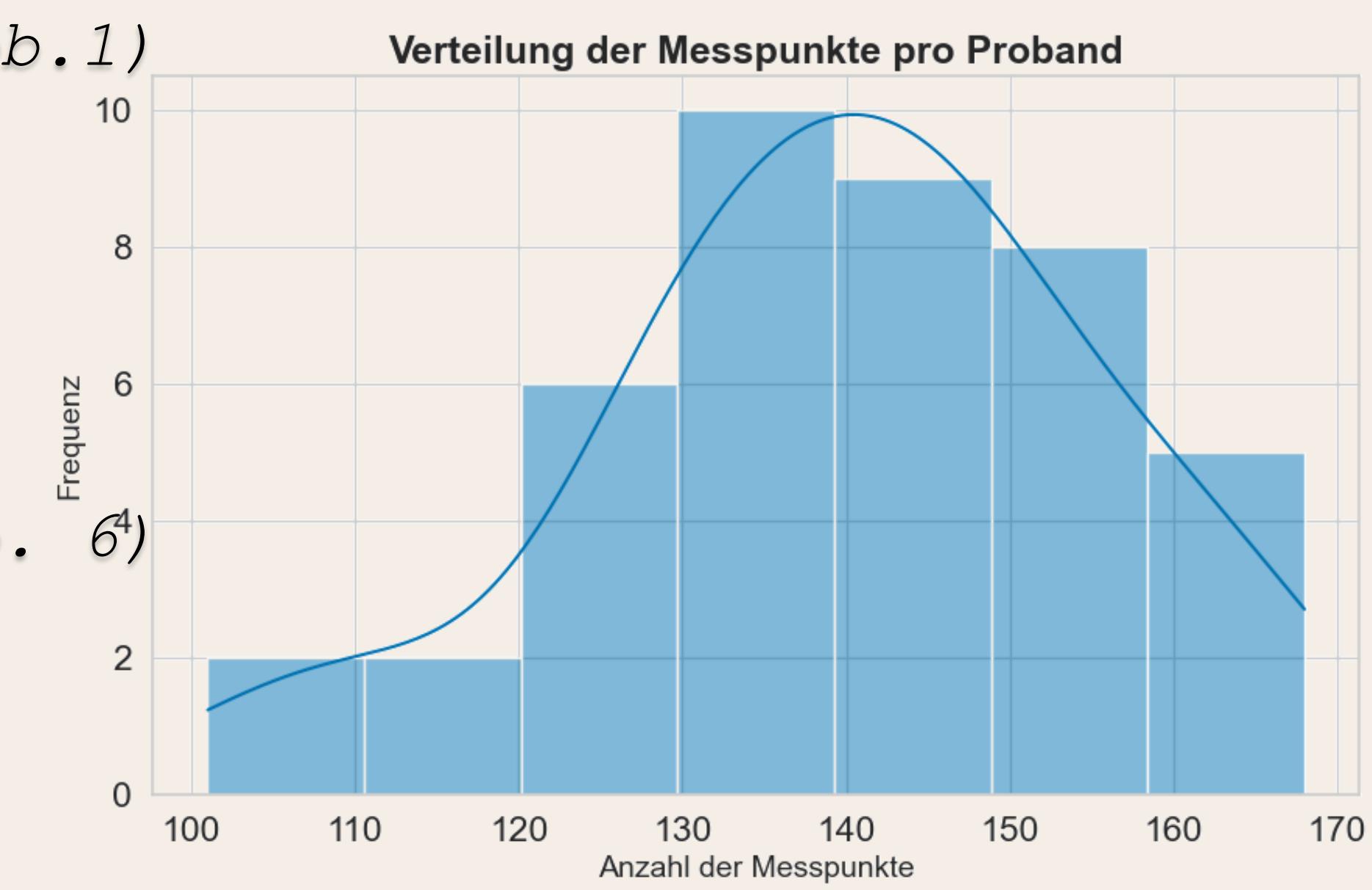


Abb. 6: Verteilung der Messpunkte pro Proband.
Quelle: Eigene Darstellung.

```
# Erstellen des Histogramms 'Verteilung der Messpunkte pro Proband' (siehe Abb. 6)
plt.figure(figsize=(10, 6))
sns.histplot(measurements_per_subject, kde=True)
plt.title('Verteilung der Messpunkte pro Proband', fontweight='bold')
plt.xlabel('Anzahl der Messpunkte')
plt.ylabel('Frequenz')
```

```
# Erstellung von Histogrammen für die Häufigkeit von motor_UPDRS und total_UPDRS Werte (siehe Abb. 7)
plt.figure(figsize=(14, 6))
```

Histogramm für motor_UPDRS

```
plt.subplot(1, 2, 1) # 1 Zeile, 2 Spalten, 1. Subplot
plt.hist(df['motor_UPDRS'], bins=30, color='skyblue', edgecolor='black')
plt.title('Häufigkeit von motor_UPDRS Werten')
plt.xlabel('motor_UPDRS')
plt.ylabel('Häufigkeit')
```

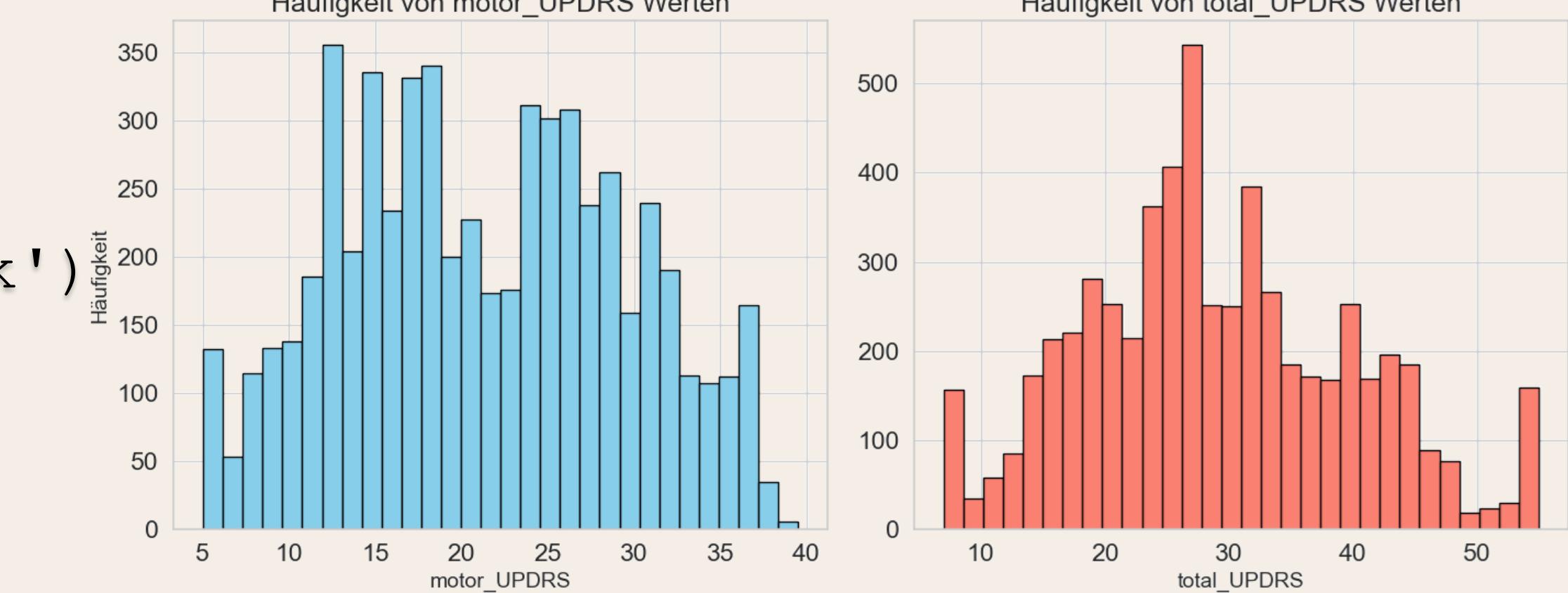


Abb. 7: Histogramm der Häufigkeit motorischer UPDRS-Werte (links) und totaler UPDRS-Werte (rechts) pro Proband. Die Diagramme visualisieren die Verteilung der UPDRS-Werte, die zur Bewertung des Parkinson-Krankheitsverlaufs genutzt werden.
Quelle: Eigene Darstellung.

Histogramm für total_UPDRS

```
plt.subplot(1, 2, 2) # 1 Zeile, 2 Spalten, 2. Subplot
plt.hist(df['total_UPDRS'], bins=30, color='salmon', edgecolor='black')
plt.title('Häufigkeit von total_UPDRS Werten')
plt.xlabel('total_UPDRS')
```

Anzeigen der Histogramme

```
plt.tight_layout()
```

```
# Korrelationsmatrix erstellen und visualisieren (siehe Abb. 8)
```

```
correlation_matrix = df.corr()
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, fmt=".2f")
plt.title('Korrelationsmatrix für den Datensatz', fontweight="bold")
```

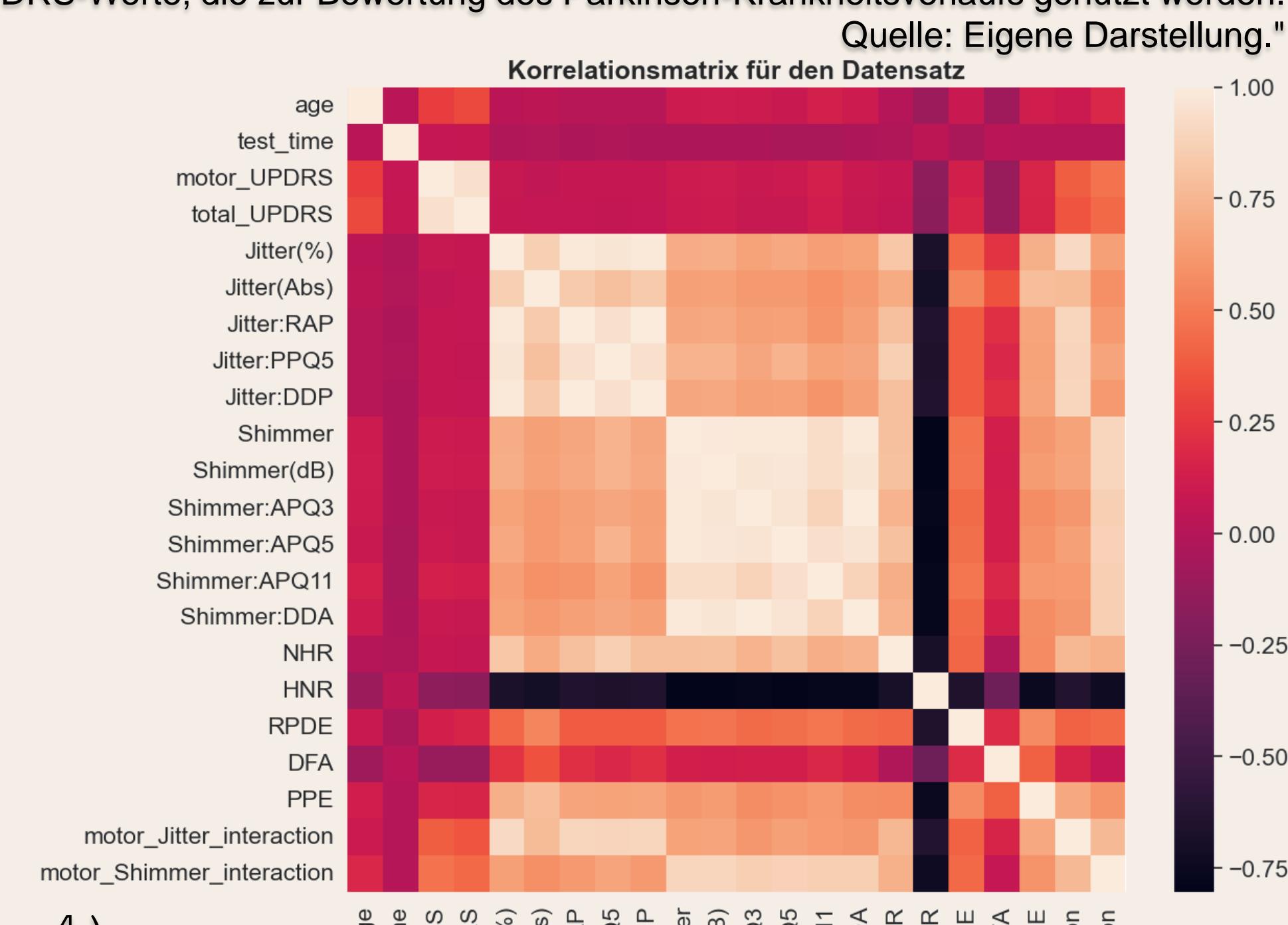


Abb. 8: Korrelationsmatrix für den Datensatz. Diese Heatmap zeigt die Korrelationen zwischen verschiedenen Stimm- und Bewegungsparametern bei Parkinson-Patienten. Dunkle Farben repräsentieren starke positive oder negative Korrelationen, während hellere Farben schwächere Korrelationen anzeigen.
Quelle: Eigene Darstellung.

```
# Erstellung Scatterplot: Vergleich diverser Messwerte (siehe Hauptposter, Abb. 4)
```

```
plt.figure(figsize=(16, 12))
plt.subplot(2, 3, 1) # Hier als Beispiel nur Subplot 1
sns.scatterplot(x="Jitter(%)", y="Shimmer", hue="sex", data=df)
plt.title('Jitter(%) vs. Shimmer', fontweight='bold')
plt.xlabel('Jitter(%) -Werte')
plt.ylabel('Shimmer -Werte')
plt.legend(title='Geschlecht')
```

```
# Histogramme für die Verteilung der Motor-Jitter und Motor-Shimmer Interaktion (siehe Hauptposter, Abb.3)
```

```
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.histplot(df['motor_Jitter_interaction'], kde=True)
plt.title('Verteilung der Motor-Jitter Interaktion')
plt.subplot(1, 2, 2)
sns.histplot(df['motor_Shimmer_interaction'], kde=True)
plt.title('Verteilung der Motor-Shimmer Interaktion')
```