Ввод [ ]:

```
# Part 1: Project Objective and Relevance
# English:
# Objective: Perform an exploratory data analysis (EDA) on an automotive dataset to uncover patterns and insights.
# Relevance: Understanding automotive data patterns is crucial for manufacturers, sellers, and buyers for informed decision-making.

# Deutsch:
# Ziel: Durchführung einer explorativen Datenanalyse (EDA) eines Automobildatensatzes zur Aufdeckung von Mustern und Erkenntnissen.
# Relevanz: Das Verständnis von Mustern in Automobildaten ist für Hersteller, Verkäufer und Käufer wichtig, um fundierte Entscheidungen tre
```

Ввод [3]:

```
# Part 2: Downloading the Dataset

import urllib3

url = "[redacted]autos.csv.bz2"

http = urllib3.PoolManager()
download = http.request("GET", url, preload_content=False)
data = download.read()

with open('autos.csv.bz2', 'wb') as f:
    f.write(data)

download.release_conn()

# English:
# This section downloads the dataset from a specified URL and saves it locally as a compressed .bz2 file.

# Deutsch:
# Dieser Abschnitt lädt den Datensatz von einer angegebenen URL herunter und speichert ihn lokal als komprimierte .bz2-Datei.
```

Ввод [4]:

```
# Part 3: Loading and Initial Cleaning of Data

import pandas as pd

df = pd.read_csv("./autos.csv.bz2", encoding = "iso8859-1")

df = df.drop("dateCrawled", axis = 1)


```

Ввод [5]:

```
print(len(df))
```

```
371528
```

Ввод [6]:

```
df.head()
```

Out[6]:

	name	seller	offerType	price	abtest	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOfRegistration	fuelType	bra
0	Golf_3_1.6	privat	Angebot	480	test	NaN	1993	manuell	0	golf	150000	0	benzin	volkswa
1	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	2011	manuell	190	NaN	125000	5	diesel	a
2	Jeep_Grand_Cherokee_"Overland"	privat	Angebot	9800	test	suv	2004	automatik	163	grand	125000	8	diesel	j
3	GOLF_4_1_4__3TÜRER	privat	Angebot	1500	test	kleinwagen	2001	manuell	75	golf	150000	6	benzin	volkswa
4	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	2008	manuell	69	fabia	90000	7	diesel	sk

Ввод [7]:

```
#print(df["abtest"].unique())

#print(len(df[df["abtest"] == "test"]))
#print(len(df[df["abtest"] == "control"]))
```

```
['test' 'control']
192585
178943
```

Ввод [ ]:

```
# English:
# The dataset is loaded into a DataFrame, and an unnecessary column (dateCrawled) is dropped.

# Deutsch:
# Der Datensatz wird in ein DataFrame geladen und eine unnötige Spalte (dateCrawled) wird entfernt.
```

Ввод [11]:

```
# Part 4: Handling Missing and Erroneous Data

import numpy as np

df["monthOfRegistration"] = np.where(df["monthOfRegistration"] == 0, 6, df["monthOfRegistration"])

df["registration"] = (df["yearOfRegistration"] + (df["monthOfRegistration"] - 1) / 12)

df = df.drop(["yearOfRegistration", "monthOfRegistration"], axis=1)
df.head()
```

Out[11]:

	name	seller	offerType	price	abtest	vehicleType	gearbox	powerPS	model	kilometer	fuelType	brand	notRepairedDamage	dateCreated	nr
0	Golf_3_1.6	privat	Angebot	480	test	NaN	manuell	0	golf	150000	benzin	volkswagen	NaN	2016-03-24 00:00:00	
1	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	manuell	190	NaN	125000	diesel	audi	ja	2016-03-24 00:00:00	
2	Jeep_Grand_Cherokee_"Overland"	privat	Angebot	9800	test	suv	automatik	163	grand	125000	diesel	jeep	NaN	2016-03-14 00:00:00	
3	GOLF_4_1_4__3TÜRER	privat	Angebot	1500	test	kleinwagen	manuell	75	golf	150000	benzin	volkswagen	nein	2016-03-17 00:00:00	
4	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	manuell	69	fabia	90000	diesel	skoda	nein	2016-03-31 00:00:00	

Ввод [ ]:

```
# English:
# Missing month of registration is replaced with 6. Registration year and month are combined into a single registration date.

# Deutsch:
# Fehlende Registrierungsmonate werden durch 6 ersetzt. Registrierungsjahr und -monat werden zu einem einzigen Registrierungsdatum kombinie
```

Ввод [12]:

```
# Part 5: Further Data Cleaning

df["seller"].unique()
```

Out[12]:

```
array(['privat', 'gewerblich'], dtype=object)
```

Ввод [13]:

```
len(df[df["seller"] == "gewerblich"])
```

Out[13]:

```
3
```

Ввод [14]:

```
len(df[df["seller"] == "privat"])
```

Out[14]:

```
371525
```

Ввод [15]:

```
df["seller"].describe()
```

Out[15]:

```
count     371528
unique         2
top       privat
freq      371525
Name: seller, dtype: object
```

Ввод [16]:

```
df.head()
```

Out[16]:

	name	seller	offerType	price	abtest	vehicleType	gearbox	powerPS	model	kilometer	fuelType	brand	notRepairedDamage	dateCreated	nr
0	Golf_3_1.6	privat	Angebot	480	test	NaN	manuell	0	golf	150000	benzin	volkswagen	NaN	2016-03-24 00:00:00	
1	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	manuell	190	NaN	125000	diesel	audi	ja	2016-03-24 00:00:00	
2	Jeep_Grand_Cherokee_"Overland"	privat	Angebot	9800	test	suv	automatik	163	grand	125000	diesel	jeep	NaN	2016-03-14 00:00:00	
3	GOLF_4_1_4__3TÜRER	privat	Angebot	1500	test	kleinwagen	manuell	75	golf	150000	benzin	volkswagen	nein	2016-03-17 00:00:00	
4	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	manuell	69	fabia	90000	diesel	skoda	nein	2016-03-31 00:00:00	

Ввод [17]:
```
df[df["price"] == 0]
```

Out[17]:

	name	seller	offerType	price	abtest	vehicleType	gearbox	powerPS	model	kilometer	fuelType	brand	notRepairedDa
7	VW_Derby_Bj_80__Scheunenfund	privat	Angebot	0	test	limousine	manuell	50	andere	40000	benzin	volkswagen	
40	Suche_Opel_corsa_a_zu_verschenken	privat	Angebot	0	test	NaN	NaN	0	corsa	150000	benzin	opel	
115	Golf_IV_1.4_16V	privat	Angebot	0	test	NaN	manuell	0	golf	5000	benzin	volkswagen	
119	Polo_6n_Karosse_zu_verschenken	privat	Angebot	0	test	kleinwagen	NaN	0	NaN	5000	benzin	volkswagen	
157	Opel_meriva_1.6_16_v_lpg__z16xe_no_OPC	privat	Angebot	0	test	bus	manuell	101	meriva	150000	lpg	opel	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
371356	Verkaufen_einen_Opel_corsa_b_worlcup_cool	privat	Angebot	0	control	NaN	manuell	65	corsa	150000	NaN	opel	
371392	Ford_Fiesta_1.3___60PS___Bj_2002___Klima___Servo	privat	Angebot	0	test	kleinwagen	manuell	60	fiesta	150000	benzin	ford	
371402	Suzuki_Swift_zu_verkaufen	privat	Angebot	0	control	kleinwagen	manuell	53	swift	150000	benzin	suzuki	
371431	Seat_Arosa	privat	Angebot	0	control	kleinwagen	manuell	37	arosa	150000	benzin	seat	
371522	Mitsubishi_Cold	privat	Angebot	0	control	NaN	manuell	0	colt	150000	benzin	mitsubishi	

10778 rows × 18 columns

Ввод [21]:
```
df = df.drop(df[df["price"] == 0].index)
df = df.drop(df[df["powerPS"] == 0].index)
```

Ввод [22]:
```
df.head()
```

Out[22]:

	name	seller	offerType	price	abtest	vehicleType	gearbox	powerPS	model	kilometer	fuelType	brand	notRepairedDamag
1	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	manuell	190	NaN	125000	diesel	audi	j
2	Jeep_Grand_Cherokee_"Overland"	privat	Angebot	9800	test	suv	automatik	163	grand	125000	diesel	jeep	Na
3	GOLF_4_1_4__3TÜRER	privat	Angebot	1500	test	kleinwagen	manuell	75	golf	150000	benzin	volkswagen	nei
4	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	manuell	69	fabia	90000	diesel	skoda	nei
5	BMW_316i___e36_Limousine___Bastlerfahrzeug__Ex...	privat	Angebot	650	test	limousine	manuell	102	3er	150000	benzin	bmw	j

Ввод [ ]:
```
# English:
# Unique sellers are identified, and records with price or powerPS equal to 0 are removed.

# Deutsch:
# Einzigartige Verkäufer werden identifiziert und Datensätze mit einem Preis oder einer Leistung (powerPS) von 0 werden entfernt.

```

Ввод [23]:
```
# Part 6: Handling Categorical Data

df["notRepairedDamage"].unique()
```

Out[23]:
```
array(['ja', nan, 'nein'], dtype=object)
```

Ввод [26]:
```
df["notRepairedDamage"]
```

Out[26]:
```
1          ja
2         NaN
3        nein
4        nein
5          ja
         ...
371520     ja
371524   nein
371525   nein
371526    NaN
371527   nein
Name: notRepairedDamage, Length: 323799, dtype: object
```

Ввод [29]:
```
df["notRepairedDamage"] = np.where(df["notRepairedDamage"] == "ja", "1", df["notRepairedDamage"])
df["notRepairedDamage"] = np.where(df["notRepairedDamage"] == "nein", "0", df["notRepairedDamage"])
```

Ввод [31]:
```
df = df[df["notRepairedDamage"].notnull()]

df.head()
```

Out[31]:

	name	seller	offerType	price	abtest	vehicleType	gearbox	powerPS	model	kilometer	fuelType	brand	notRepairedDamag
1	A5_Sportback_2.7_Tdi	privat	Angebot	18300	test	coupe	manuell	190	NaN	125000	diesel	audi	
3	GOLF_4_1_4__3TÜRER	privat	Angebot	1500	test	kleinwagen	manuell	75	golf	150000	benzin	volkswagen	
4	Skoda_Fabia_1.4_TDI_PD_Classic	privat	Angebot	3600	test	kleinwagen	manuell	69	fabia	90000	diesel	skoda	
5	BMW_316i___e36_Limousine___Bastlerfahrzeug__Ex...	privat	Angebot	650	test	limousine	manuell	102	3er	150000	benzin	bmw	
6	Peugeot_206_CC_110_Platinum	privat	Angebot	2200	test	cabrio	manuell	109	2_reihe	150000	benzin	peugeot	

Ввод [ ]:
```
# English:
# Categorical data in notRepairedDamage is converted from "ja"/"nein" to "1"/"0". Null values are removed.

# Deutsch:
# Kategoriale Daten in notRepairedDamage werden von "ja"/"nein" in "1"/"0" umgewandelt. Nullwerte werden entfernt.

```

Ввод [28]:
```
# Part 7: Filtering Data

import math

print(math.nan == math.nan)
```

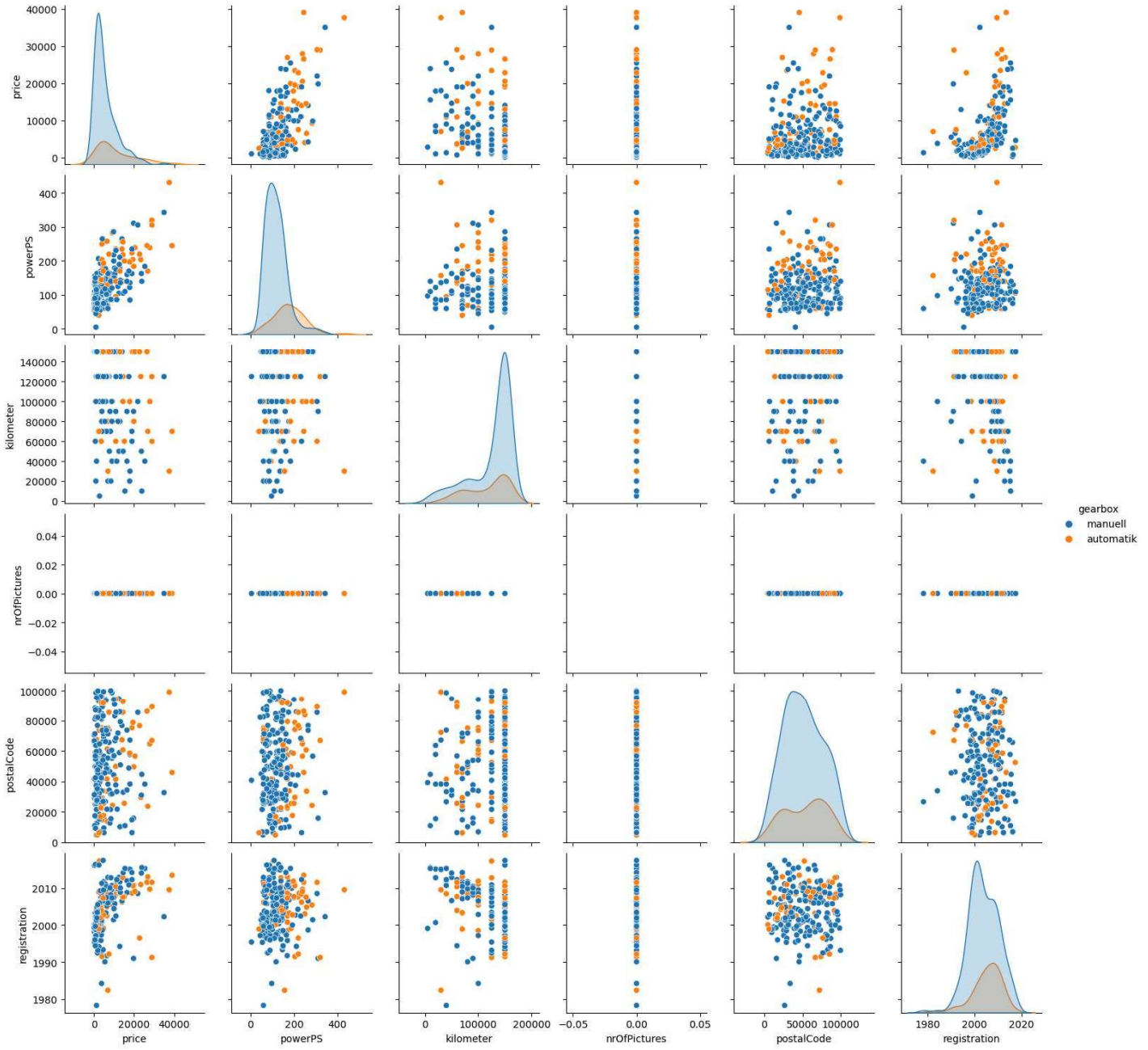
```
False
```

Ввод [40]:
```
df = df[(df["price"] < 50000) & (df["powerPS"] <500) & (df["registration"] <= 2018)]
```

Ввод [ ]:
```
# English:
# Data is filtered to exclude outliers: prices above 50,000, powerPS above 500, and registration dates after 2018.

# Deutsch:
# Daten werden gefiltert, um Ausreißer auszuschließen: Preise über 50.000, powerPS über 500 und Registrierungsdaten nach 2018.
```

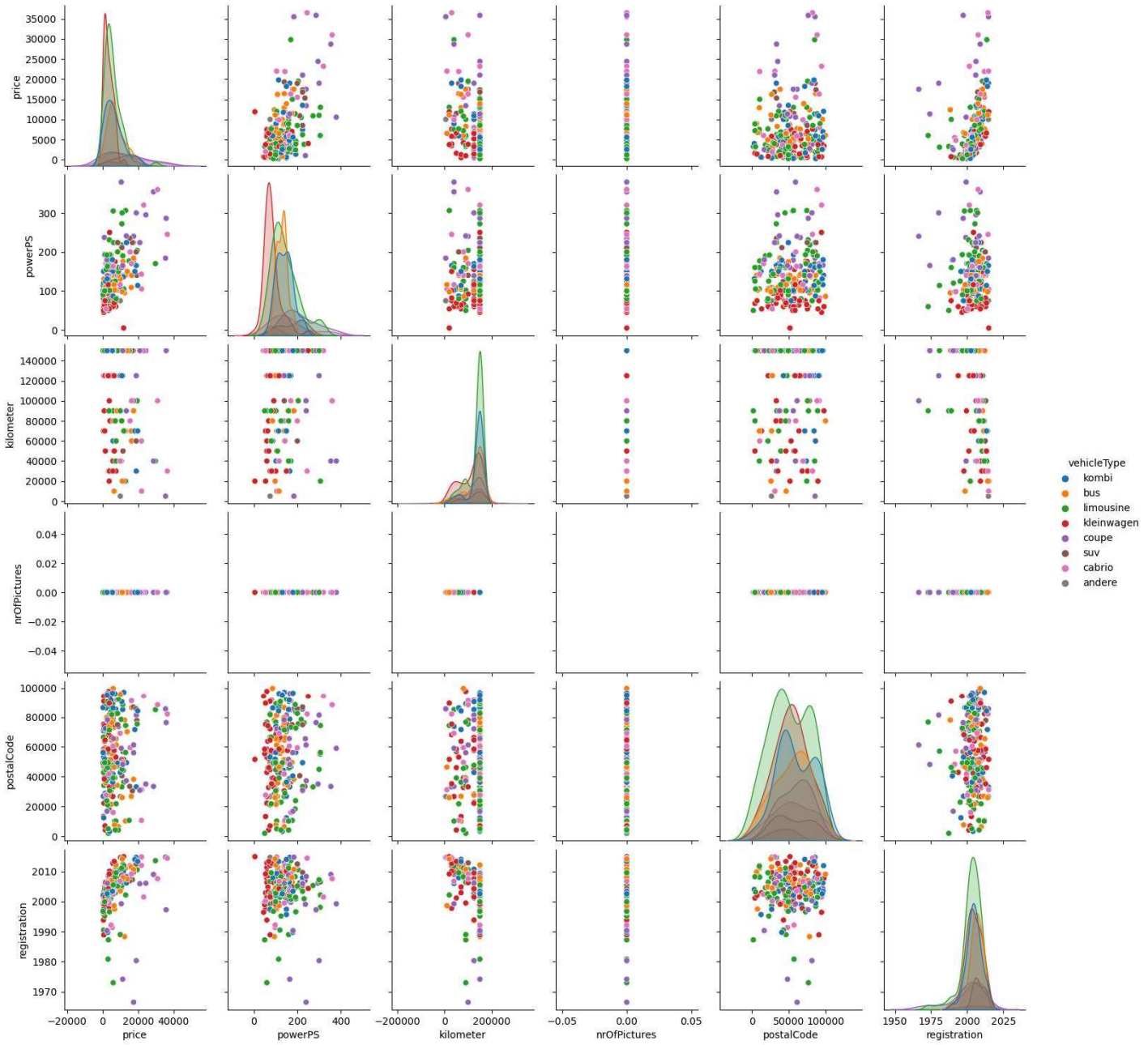
Ввод [41]:

```
# Part 8: Visualization

import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt

g = sns.pairplot(df.sample(250), hue="gearbox")
plt.show()
```



Ввод [44]:

```
g = sns.pairplot(df.sample(250), hue="vehicleType")
```



Ввод [ ]:

```
# English:
# The seaborn library is used to create pair plots to visualize relationships between features, colored by gearbox type and vehicle type.

# Deutsch:
# Die Bibliothek seaborn wird verwendet, um Paar-Diagramme zu erstellen, die Beziehungen zwischen Merkmalen darstellen, gefärbt nach Getrie

```