xsesta07 xkulis03 xpleva07

November 22, 2022

1 Příprava dat a jejich popisná charakteristika

1.1 Autoři:

- Vojtěch Kulíšek
- Lukáš Plevač
- Pavel Šesták

1.2 Zadání

Z dostupných datových sad si zvolte jednu datovou sadu, kterou se budete dále zabývat. Stáhněte si zvolenou datovou sadu z uvedeného zdroje a prostudujte si dostupné informace k této datové sadě. Proveďte explorativní analýzu zvolené datové sady. Pro každý následující bod implementujte odpovídající sekci ve zdrojovém kódu a zjištěné výsledky popište v dokumentaci: prozkoumejte jednotlivé atributy datové sady, jejich typ a hodnoty, kterých nabývají (počet hodnot, nejčastější hodnoty, rozsah hodnot atd.) prozkoumejte rozložení hodnot jednotlivých atributů pomocí vhodných grafů, zaměřte se i na to, jak hodnota jednoho či dvou atributů ovlivní rozložení hodnot jiného atributu. Do dokumentace vložte alespoň 5 různých grafů, zobrazujících zjištěná rozložení hodnot. Použijte různé typy grafů (např. bodový graf, histogram, krabicový nebo houslový graf, graf složený z více podgrafů apod.). zjistěte, zda zvolená datová sada obsahuje nějaké odlehlé hodnoty. proveďte podrobnou analýzu chybějící hodnot (celkový počet chybějících hodnot, počet objektů s více chybějícími hodnotami atd.). proveďte korelační analýzu numerických atributů (k analýze využijte i grafy a korelační koeficienty). Připravte 2 varianty datové sady vhodné pro dolovací algoritmy. Můžete uvažovat dolovací úlohu uvedenou u datové sady nebo navrhnout vlastní dolovací úlohy. V případě vlastní dolovací úlohy ji specifikujte v dokumentaci. V rámci přípravy datové sady proveďte následující kroky: Odstraňte z datové sady atributy, které jsou pro danou dolovací úlohu irelevantní. Vypořádejte se s chybějícími hodnotami. Pro odstranění těchto hodnot využijte alespoň dvě různé metody pro odstranění chybějících hodnot. Vypořádejte se s odlehlými hodnotami, jsou-li v datové sadě přítomny. Pro jednu variantu datové sady proveďte diskretizaci numerických atributů tak, aby výsledná datová sada byla vhodná pro algoritmy, které vyžadují na vstupu kategorické atributy. Pro druhou variantu datové sady proveďte vhodnou transformaci kategorických atributů na numerické atributy. Dále pak proveďte normalizaci numerických atributů, které má smysl normalizovat. Výsledná datová sada by měla být vhodná pro metody vyžadující numerické vstupy.

[1]:
```
import subprocess
import sys
import os
```

```
requirementsPath = os.path.join(os.path.dirname(os.path.
 ↪realpath('__file__')),"requirements.txt")
subprocess.check_call([sys.executable, "-m", "pip", "install", "-r",␣
 ↪requirementsPath])
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
import pandas as pd
from sklearn.linear_model import LinearRegression
import re
from scipy import stats

import warnings
warnings.filterwarnings('ignore')

PLOT_GRAPHS = True
PLOT_STATS = True
```

1.3 Explorační analýza

V této části se blíže seznámíme s daty, které máme dále upravovat. V rámcí modelu CRISP-DM jsme v sekci pochopení dat. Pro tuto úlohu jsme si zvolili datovou sadu z průzkumu platů v IT sektoru z roků 2018 až 2020. Dále budeme pracovat pouze s nejnovějšími daty z roku 2020 jelikož ekonomická situace je v dnešní době velmi dynamická a už tak se jedná o stará data. Starší data můžeme dále použít pro validaci našich klasifikátorů a porovnat jak moc se datové sady vzájemně liší.

1.3.1 Načtení datových souborů

V této sekci si nahrajeme zvolený datový soubor do operační paměti pomocí knihovny pandas. Z datové odstraníme atributy, které zjevně nepocházejí od uživatele jako je například časová značka. Jelikož jsou to data z dotazníku, tak některé otázky jsou rozsáhle popsány, aby uživatel věděl co přesně má vyplnit, pro naše účely si tyto sloupce přejmenujeme, aby se s daty dále lépe pracovalo.

[2]:
```
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)


FILES = ["data/IT_Salary_Survey_EU_2018.csv", "data/IT_Salary_Survey_EU_2019.
 ↪csv", "data/IT_Salary_Survey EU_2020.csv"]

data = pd.read_csv(FILES[2])

if "Timestamp" in data:
    data.drop(["Timestamp"], axis=1, inplace=True)

if "Zeitstempel" in data:
    data.drop(["Zeitstempel"], axis=1, inplace=True)
```

```
if "0" in data:
    data.drop(["0"], axis=1, inplace=True)

data.rename(columns = {
    "Your main technology / programming language":'Main Technology',
    "Other technologies/programming languages you use often" : "Other␣
 ↪technologies",
    "Yearly brutto salary (without bonus and stocks) in EUR": "Yearly brutto",
    "Annual bonus+stocks one year ago. Only answer if staying in same country" :
 ↪ "Bonus and stocks in same country",
    "Have you lost your job due to the coronavirus outbreak?" : "Job lost due␣
 ↪covid",
    "Have you received additional monetary support from your employer due to␣
 ↪Work From Home? If yes, how much in 2020 in EUR" : "Home office␣
 ↪compensation",
    "Position " : "Position",
    }, inplace = True)

data.sample(10)
```

[2]:

	Age	Gender	City	Position	\
219	35.0	Male	Berlin	Backend Developer	
828	30.0	Male	Hannover	Software Engineer	
23	59.0	Male	Berlin	Backend Developer	
613	37.0	Male	Berlin	Embedded Software Engineer	
1093	27.0	Male	Frankfurt	IT Manager	
794	39.0	Male	Berlin	Frontend Developer	
1140	26.0	Female	Berlin	Mobile Developer	
524	29.0	Female	Berlin	Software Developer in Test	
467	36.0	Male	Munich	IT Spezialist	
752	33.0	Male	Heidelberg	Software Engineer	

	Total years of experience	Years of experience in Germany	Seniority level	\
219	6	2	Senior	
828	5	3	Middle	
23	30	30	Senior	
613	15	1	Senior	
1093	5	3	Middle	
794	10	3	Senior	
1140	8	3	Middle	
524	9	4	Lead	
467	15	6	Senior	
752	11	5	Senior	

	Main Technology	Other technologies	\
219	Java, angular, Aws	Python, Javascript / Typescript, Java / Scala,…	

828	Python	Javascript / Typescript, Java / Scala, SQL
23	Java	Kotlin, Java / Scala, SQL
613	C/C++	Python, C/C++
1093	NaN	NaN
794	Javascript	Javascript / Typescript, Ruby, AWS, Docker
1140	c++	C/C++, SAP / ABAP
524	Java	Javascript / Typescript, SQL, Docker
467	AWS	Javascript / Typescript, Java / Scala, AWS, Do…
752	Python	Python, Javascript / Typescript, SQL, AWS, Kub…

	Yearly brutto	Yearly bonus + stocks in EUR
219	74000.0	2000
828	45000.0	0
23	69000.0	NaN
613	78000.0	0
1093	73000.0	4000
794	85000.0	NaN
1140	40000.0	2000
524	66000.0	0
467	79300.0	11900
752	80000.0	0

	Annual brutto salary (without bonus and stocks) one year ago. Only answer if staying in the same country
219	NaN
828	45000.0
23	69000.0
613	78000.0
1093	65000.0
794	NaN
1140	15000.0
524	60000.0
467	73000.0
752	NaN

	Bonus and stocks in same country	Number of vacation days
219	NaN	30
828	0	30
23	2000	28
613	0	28
1093	3000	37
794	NaN	27
1140	NaN	36
524	0	25
467	6000	30
752	78000	30

	Employment status	ontract duration	Main language at work
219	Full-time employee	Unlimited contract	English
828	Full-time employee	Unlimited contract	English
23	Full-time employee	Unlimited contract	German
613	Full-time employee	Unlimited contract	English
1093	Full-time employee	Unlimited contract	English
794	Full-time employee	Unlimited contract	English
1140	Self-employed (freelancer)	Temporary contract	German
524	Full-time employee	Unlimited contract	English
467	Full-time employee	Unlimited contract	German
752	Full-time employee	Temporary contract	English

	Company size	Company type	Job lost due covid
219	51-100	Product	No
828	up to 10	Startup	No
23	101-1000	Product	No
613	1000+	Product	No
1093	1000+	Industry	No
794	101-1000	Product	No
1140	101-1000	Product	No
524	101-1000	Startup	No
467	1000+	Product	No
752	101-1000	Research institute	No

	Have you been forced to have a shorter working week (Kurzarbeit)? If yes, how many hours per week
219	32.0
828	8.0
23	0.0
613	0.0
1093	NaN
794	NaN
1140	NaN
524	NaN
467	0.0
752	NaN

	Home office compensation
219	NaN
828	0
23	NaN
613	NaN
1093	150
794	NaN
1140	NaN
524	NaN
467	NaN

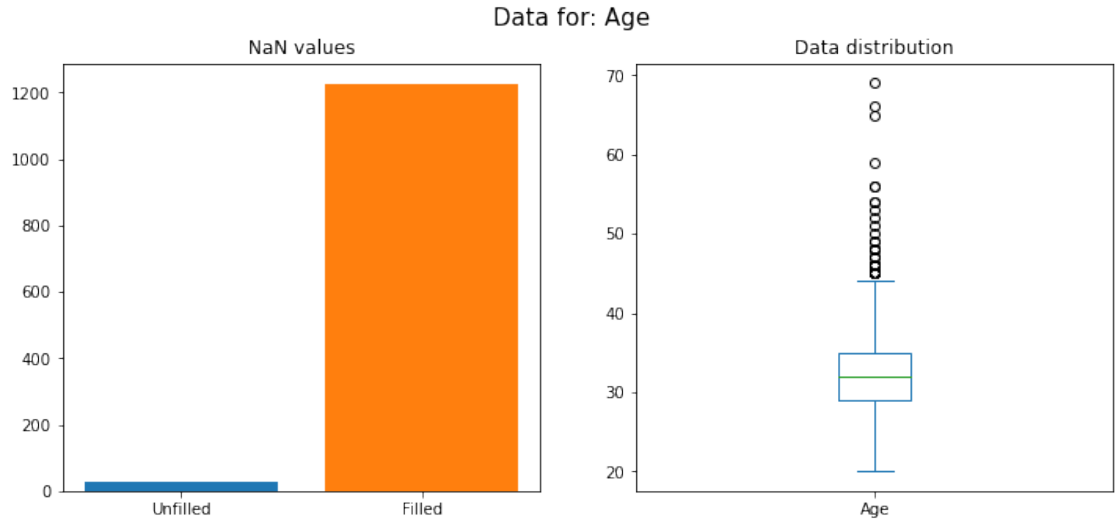
```
752                          NaN
```

[3]:
```
#pd.plotting.parallel_coordinates(data, "Your level")
#plt.show()
#note1 i found whole line with NaN filter it
```

[4]:
```
def plot_graphs(data: pd.DataFrame) -> None:
    """
    Get pandas dataframe. Describe and plot graphs for all columns in dataframe.

    PRE CONDITION: If you want just text info about params specify PLOT_STATS.␣
 ↪If you want pyplot graphs as output define constant PLOT_GRAPHS to True.
    """
    if not PLOT_STATS and not PLOT_GRAPHS:
        return
    columns = data.columns

    dtypes = data.dtypes

    for column in columns:

        print(data[column].describe())

        if not PLOT_GRAPHS:
            continue

        figure, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 5))
        figure.suptitle("Data for: "+ column, fontsize=15)
        axes[0].set_title("NaN values", fontsize=12)

        IsNan = data[column].isna().sum()
        IsNotNan = len(data[column])-IsNan
        axes[0].bar("Unfilled", IsNan)
        axes[0].bar("Filled", IsNotNan)
        axes[1].set_title("Data distribution", fontsize=12)

        if dtypes[column] == "object":
            data[column].value_counts().plot(kind='bar')
        elif dtypes[column] == "float64":
            data[column].plot(kind='box')

        plt.show()

plot_graphs(data)
```

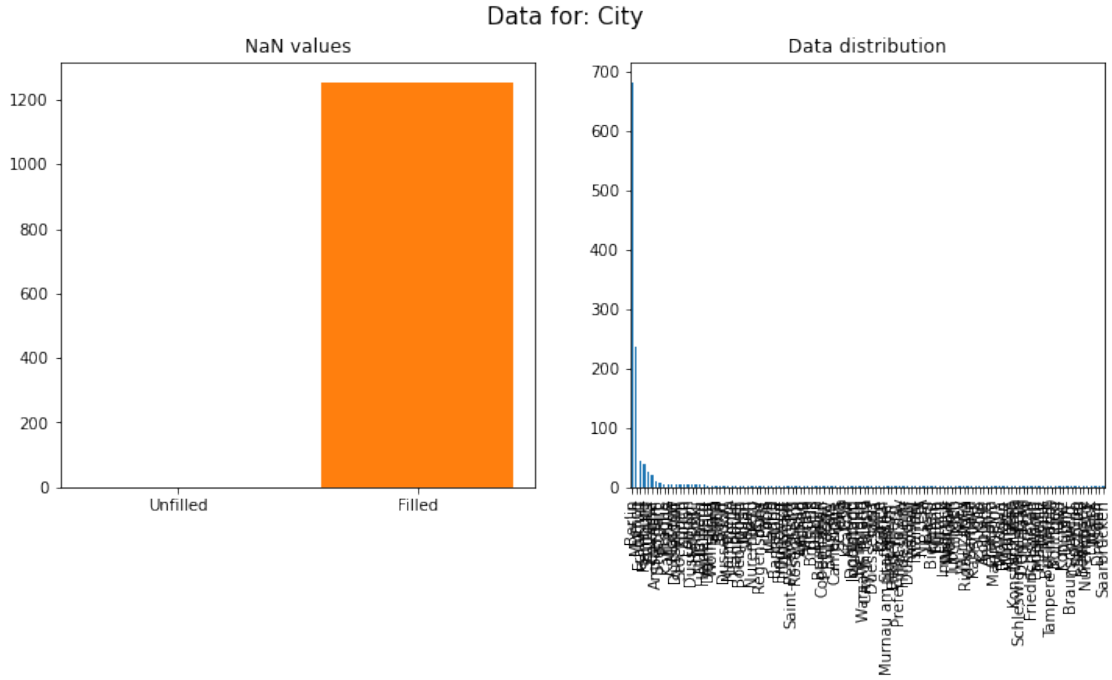
```
count    1226.000000
mean       32.509788
std         5.663804
```

```
min         20.000000
25%         29.000000
50%         32.000000
75%         35.000000
max         69.000000
Name: Age, dtype: float64
```

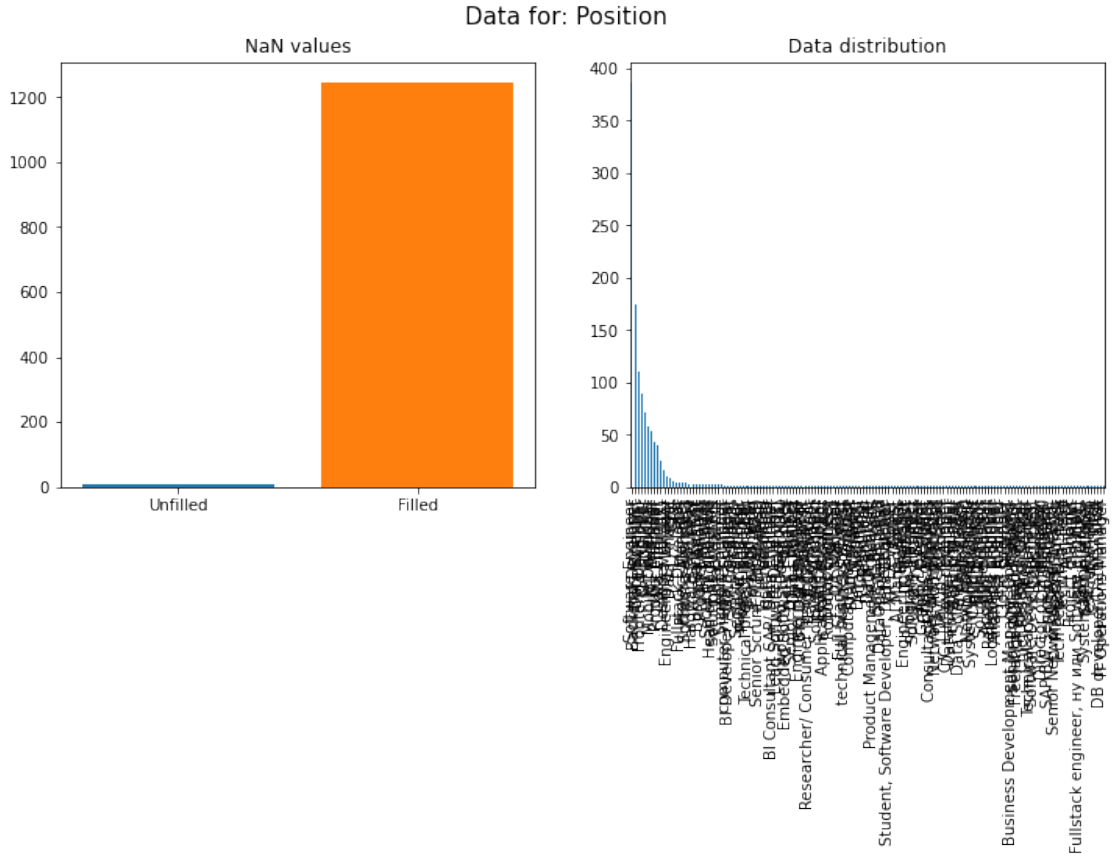



```
count     1243
unique       3
top       Male
freq      1049
Name: Gender, dtype: object
```

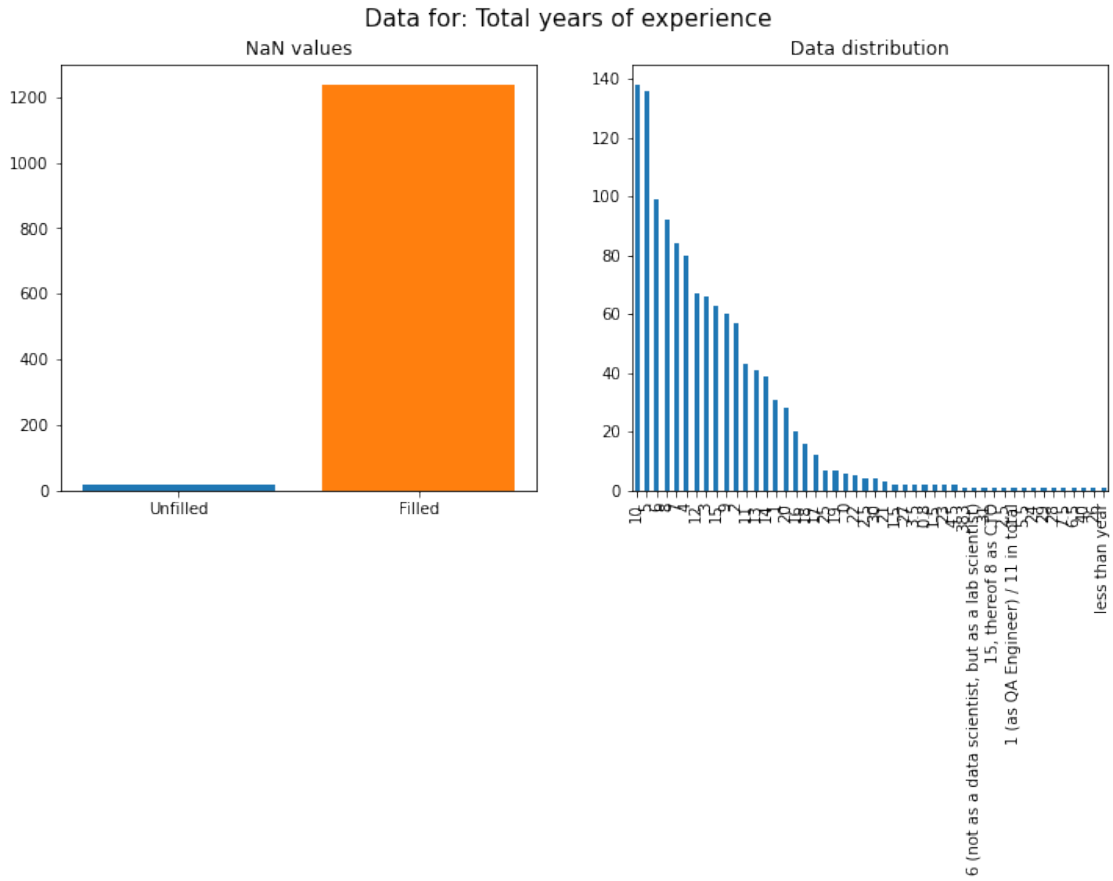
```
count       1253
unique       119
top       Berlin
freq         681
Name: City, dtype: object
```



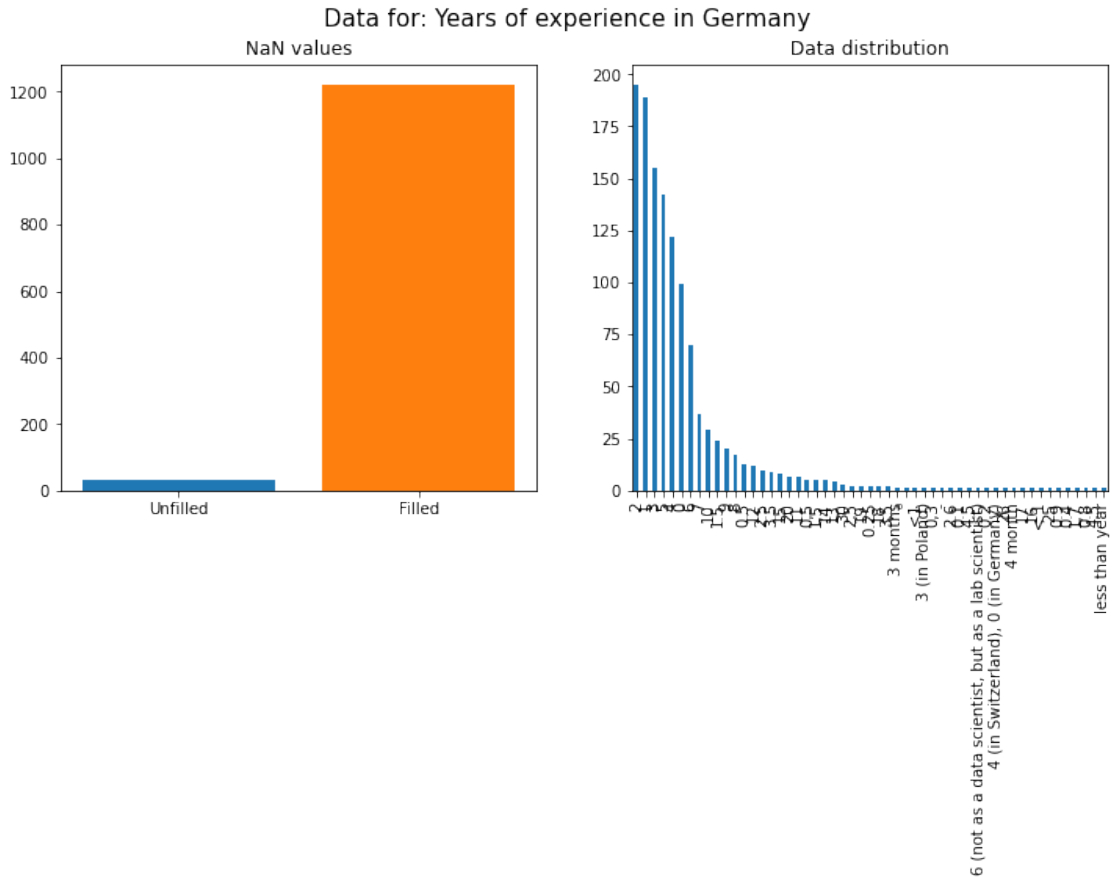

```
count                  1247
unique                  148
top       Software Engineer
freq                    387
Name: Position, dtype: object
```



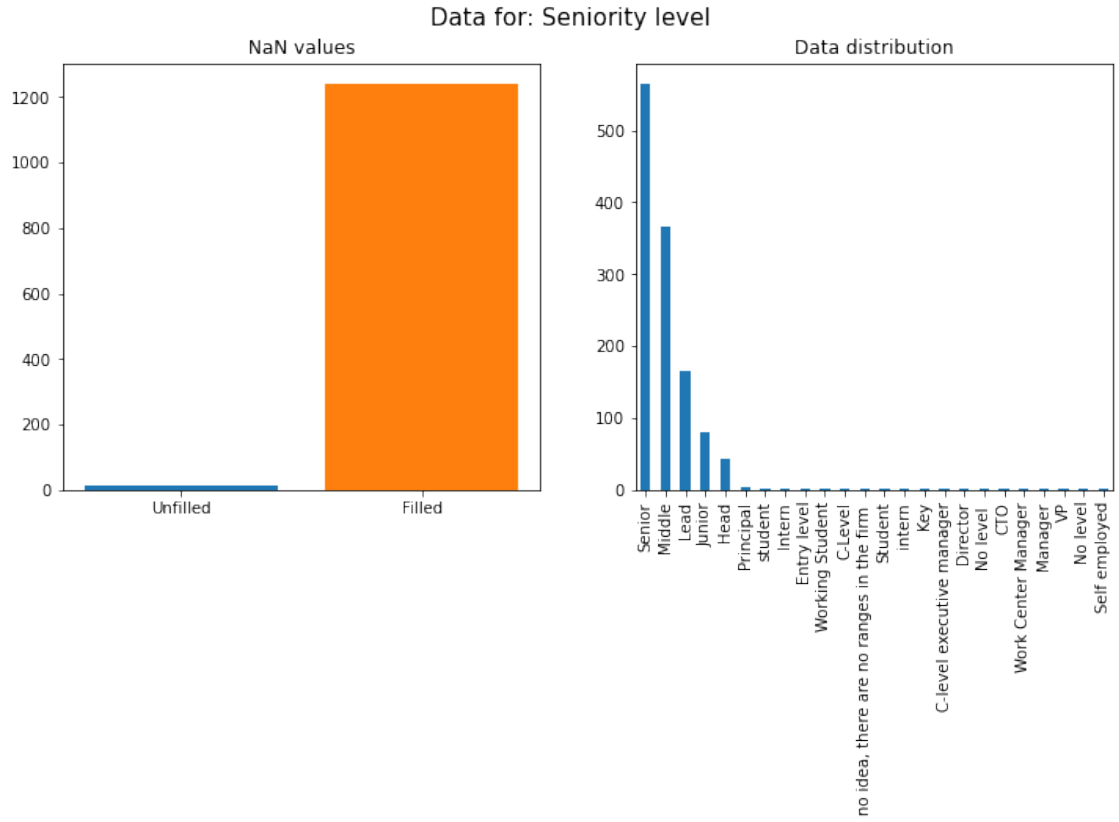

```
count     1237
unique      48
top         10
freq       138
Name: Total years of experience, dtype: object
```



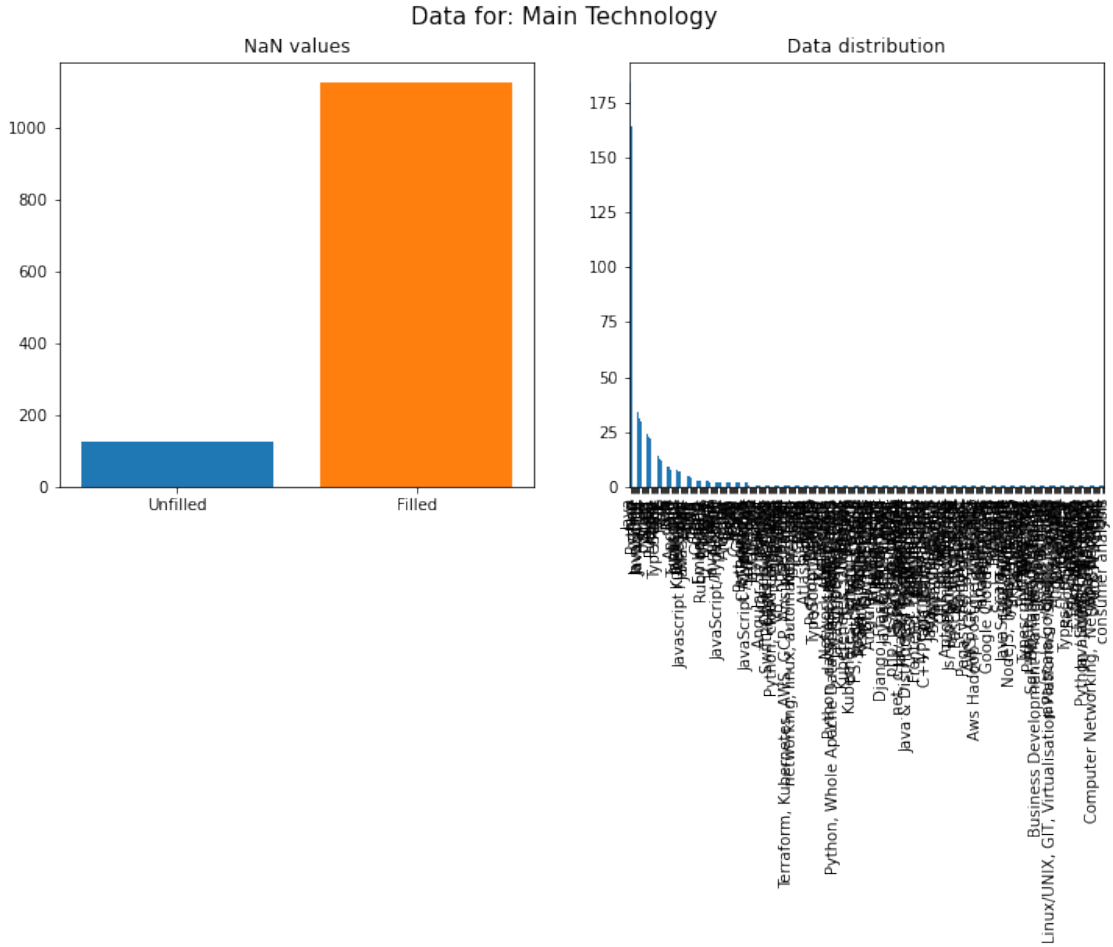

```
count     1221
unique      53
top          2
freq       195
Name: Years of experience in Germany, dtype: object
```



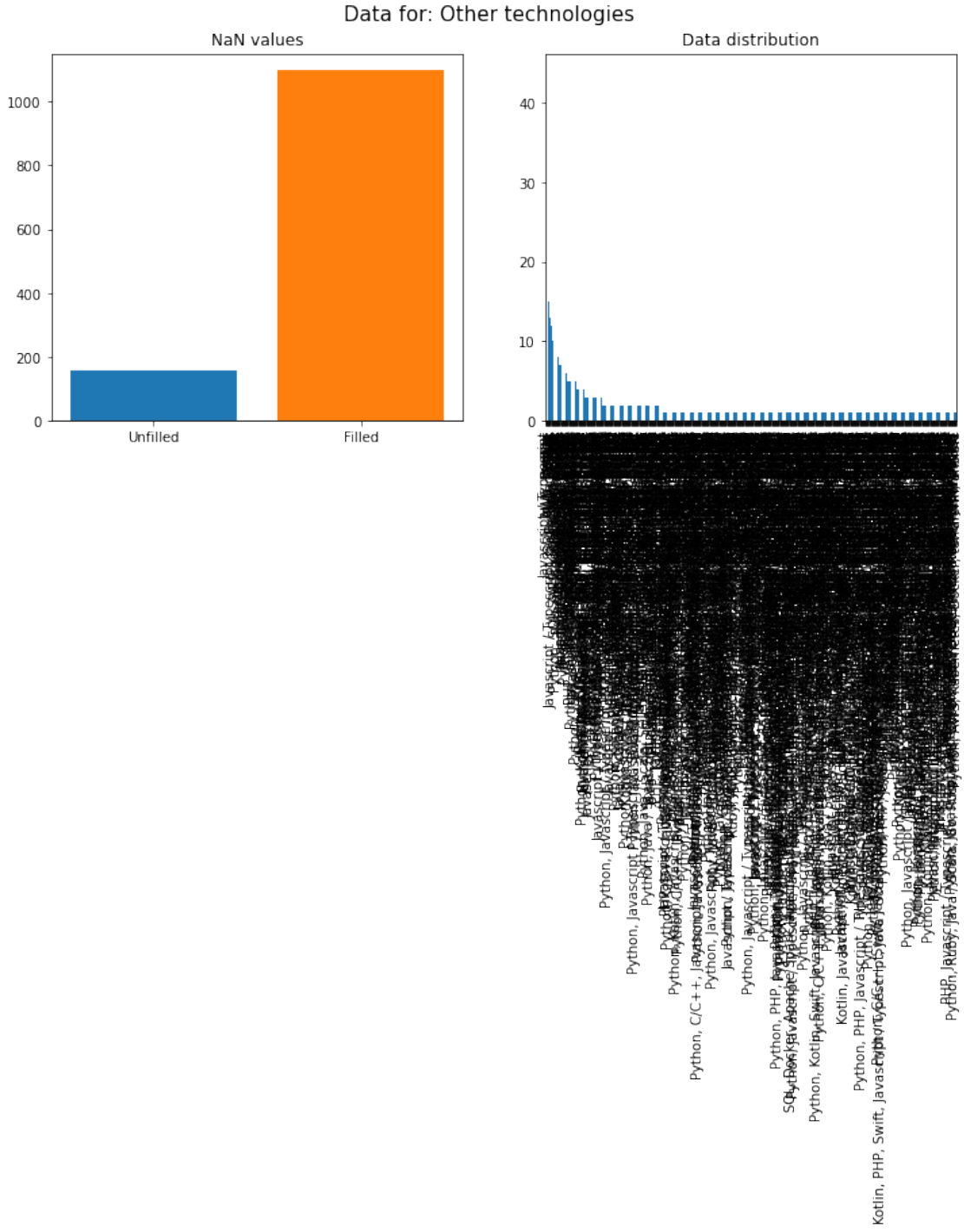

```
count       1241
unique        24
top       Senior
freq         565
Name: Seniority level, dtype: object
```




```
count     1126
unique     256
top       Java
freq       184
Name: Main Technology, dtype: object
```

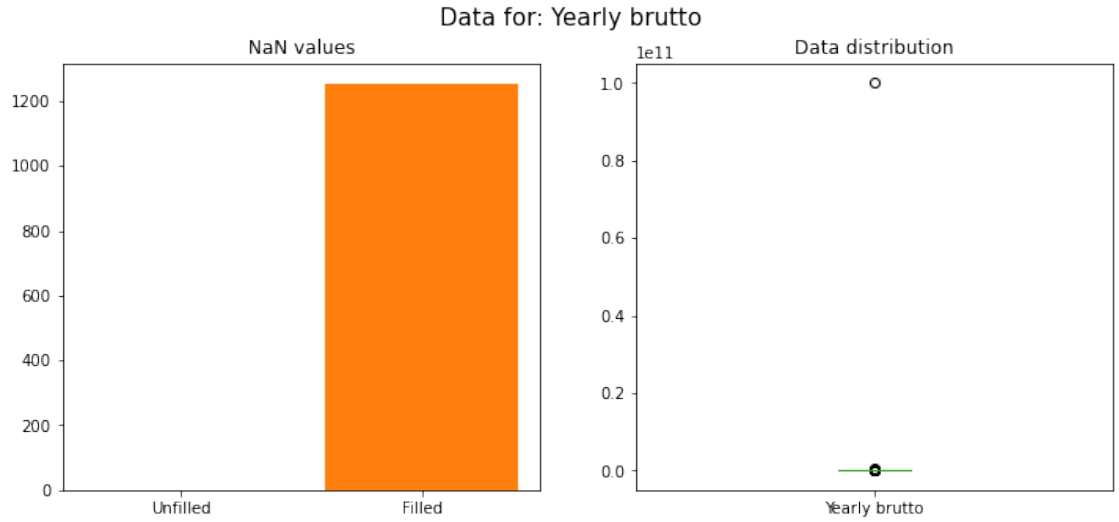



```
count                          1096
unique                          562
top       Javascript / Typescript
freq                             44
Name: Other technologies, dtype: object
```

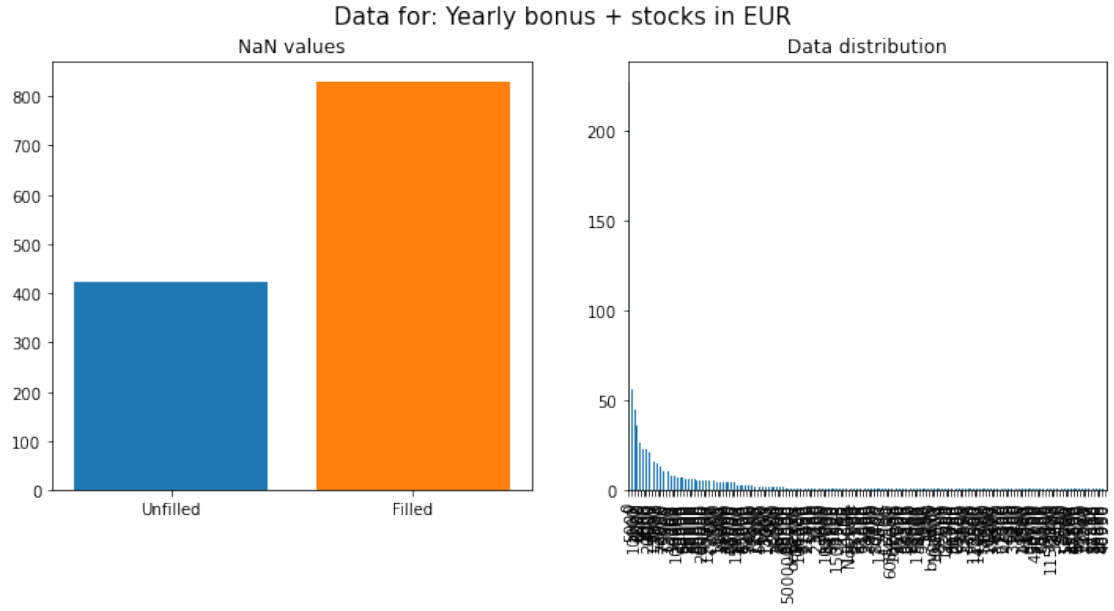


count	1.253000e+03
mean	8.027904e+07
std	2.825061e+09
min	1.000100e+04
25%	5.880000e+04

```
50%       7.000000e+04
75%       8.000000e+04
max       1.000000e+11
Name: Yearly brutto, dtype: float64
```

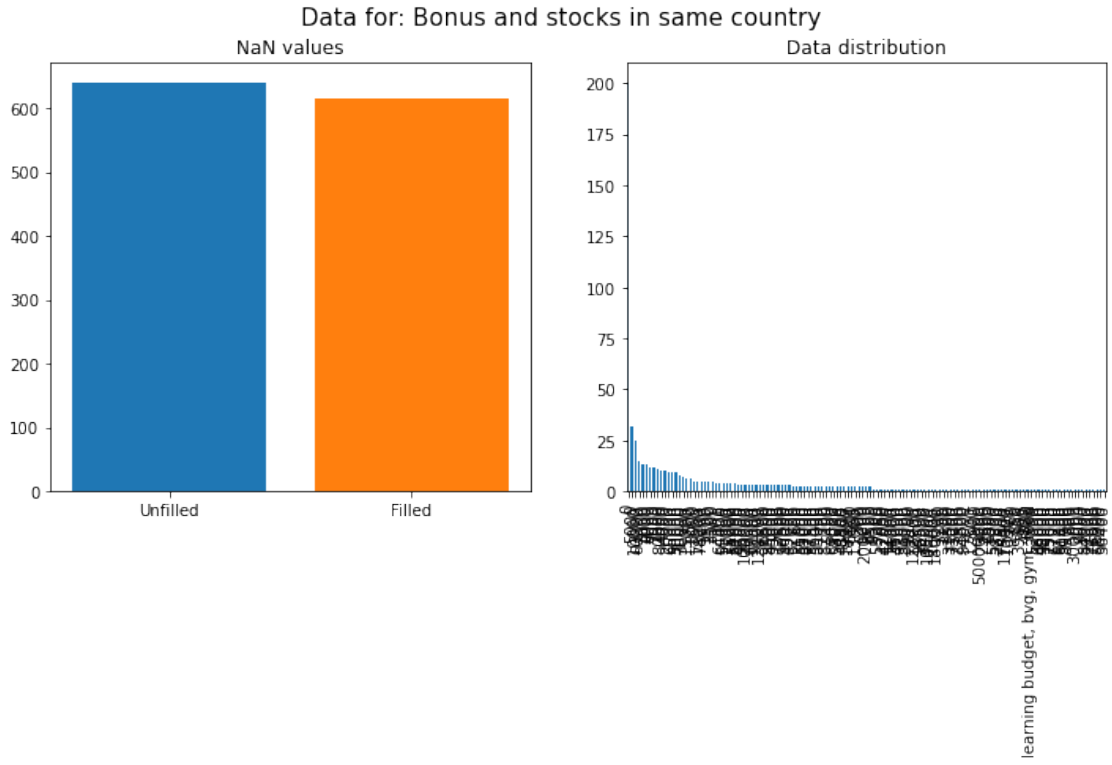



```
count      829
unique     168
top          0
freq       227
Name: Yearly bonus + stocks in EUR, dtype: object
```

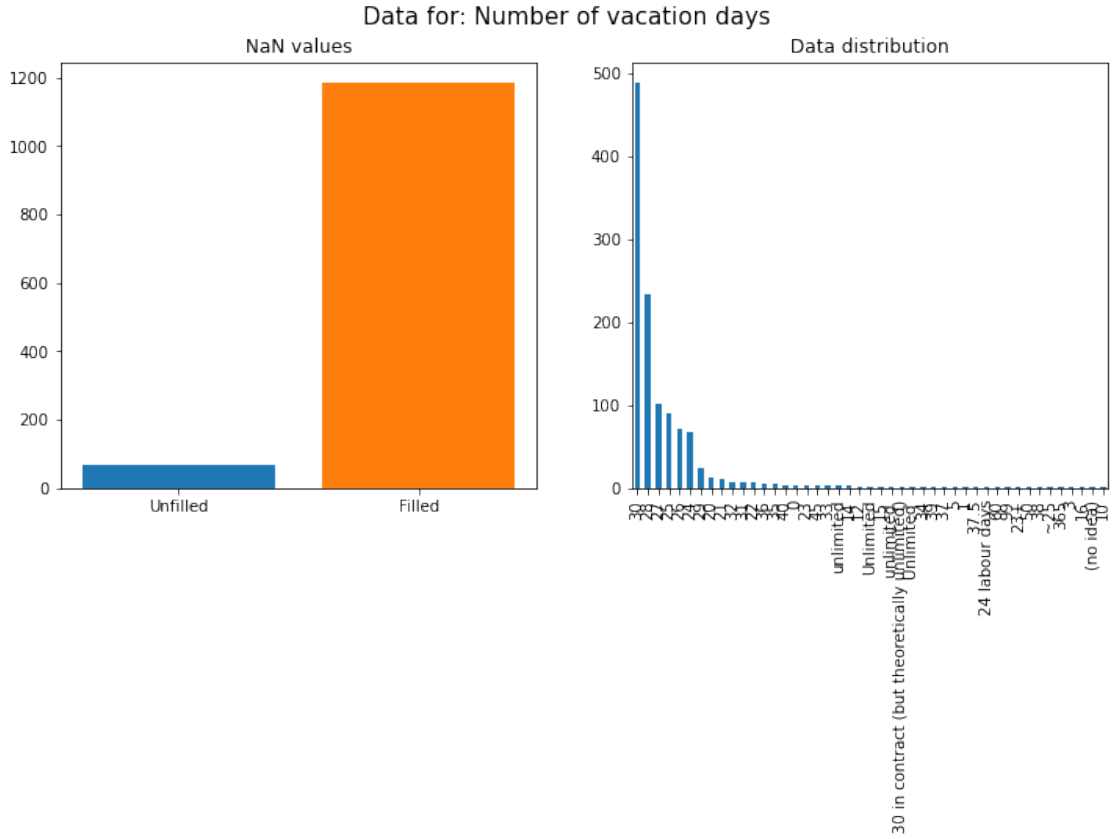


```
count    8.850000e+02
mean     6.322459e+05
std      1.680508e+07
min      1.100000e+04
25%      5.500000e+04
50%      6.500000e+04
75%      7.500000e+04
max      5.000000e+08
Name: Annual brutto salary (without bonus and stocks) one year ago. Only answer
if staying in the same country, dtype: float64
```

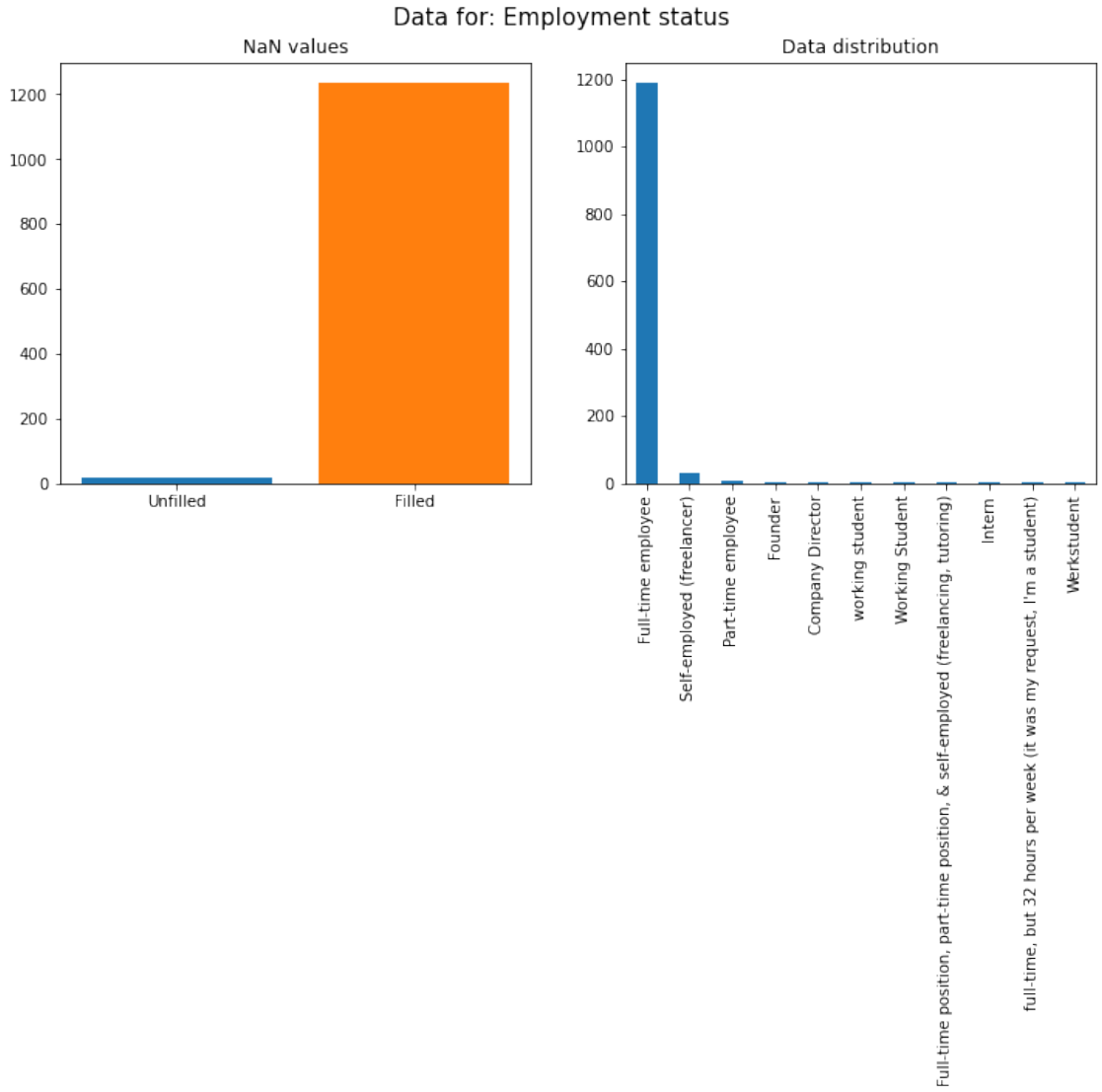

```
count      614
unique     131
top          0
freq       200
Name: Bonus and stocks in same country, dtype: object
```



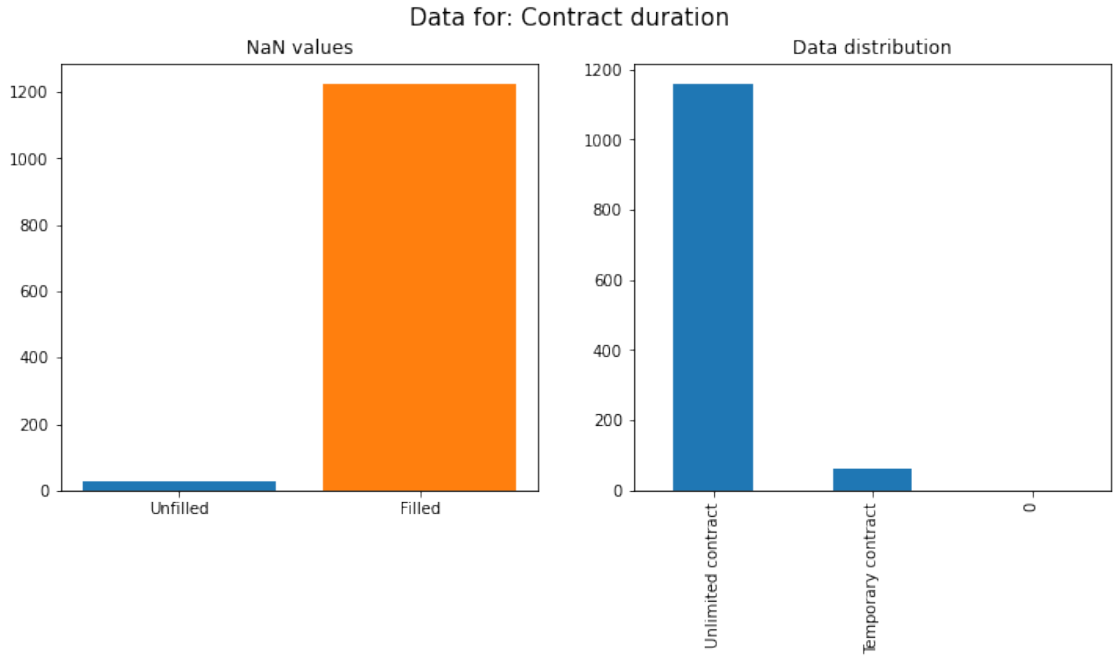

```
count     1185
unique      45
top         30
freq       488
Name: Number of vacation days, dtype: object
```



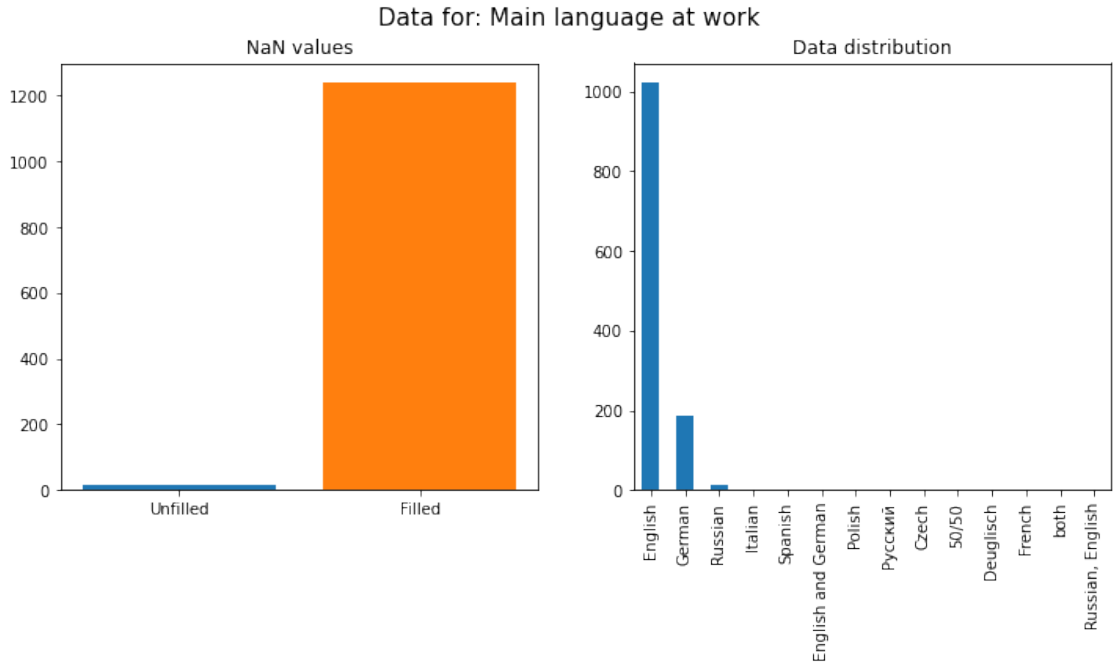

```
count                   1236
unique                    11
top       Full-time employee
freq                    1190
Name: Employment status, dtype: object
```



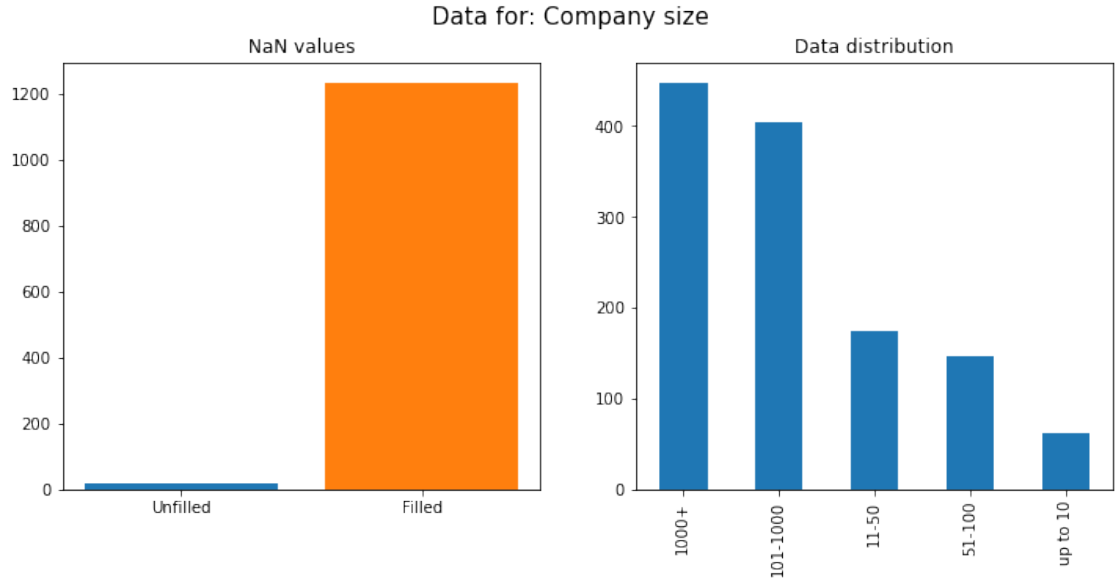

```
count                   1224
unique                     3
top       Unlimited contract
freq                    1159
Name:  ontract duration, dtype: object
```



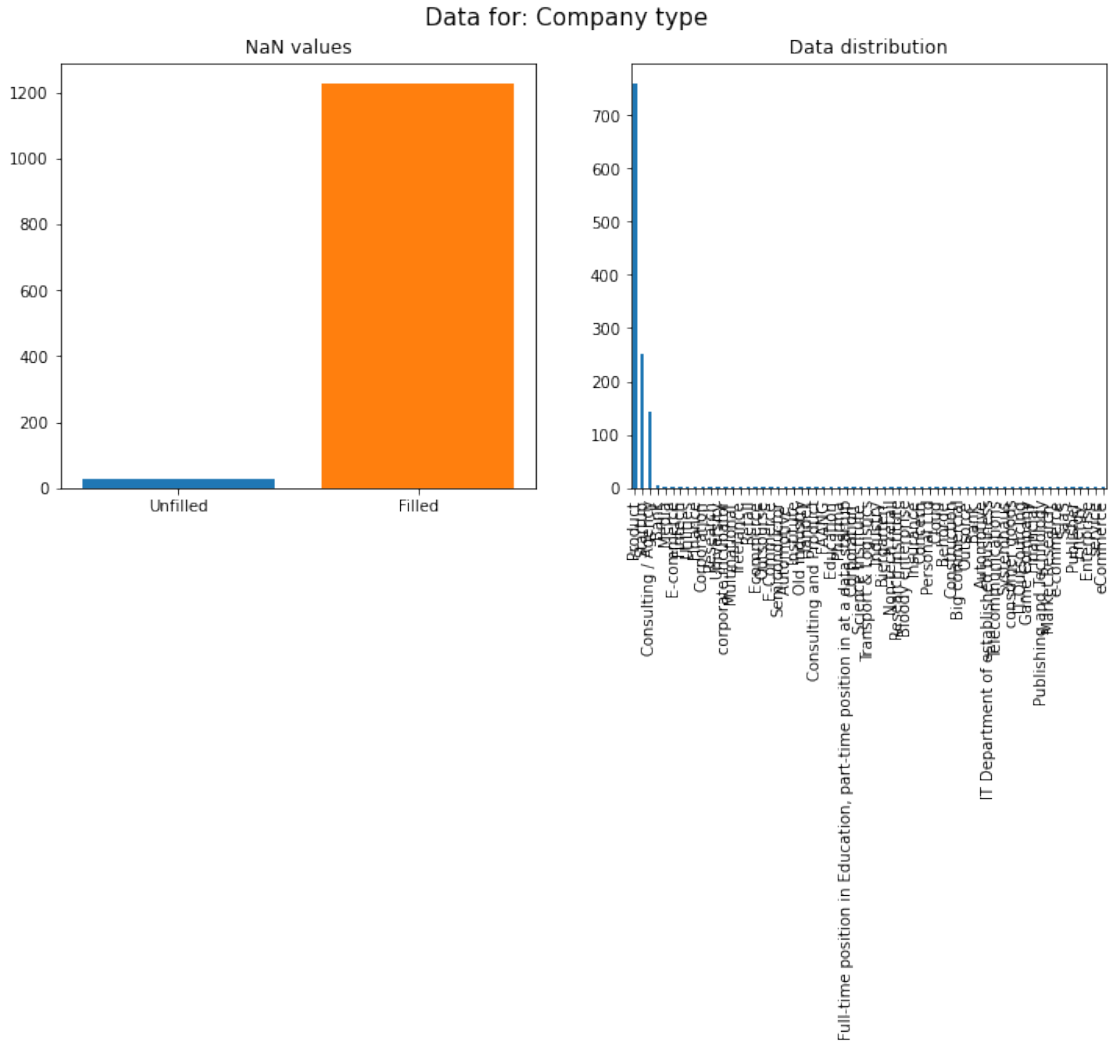

```
count         1237
unique          14
top        English
freq          1020
Name: Main language at work, dtype: object
```



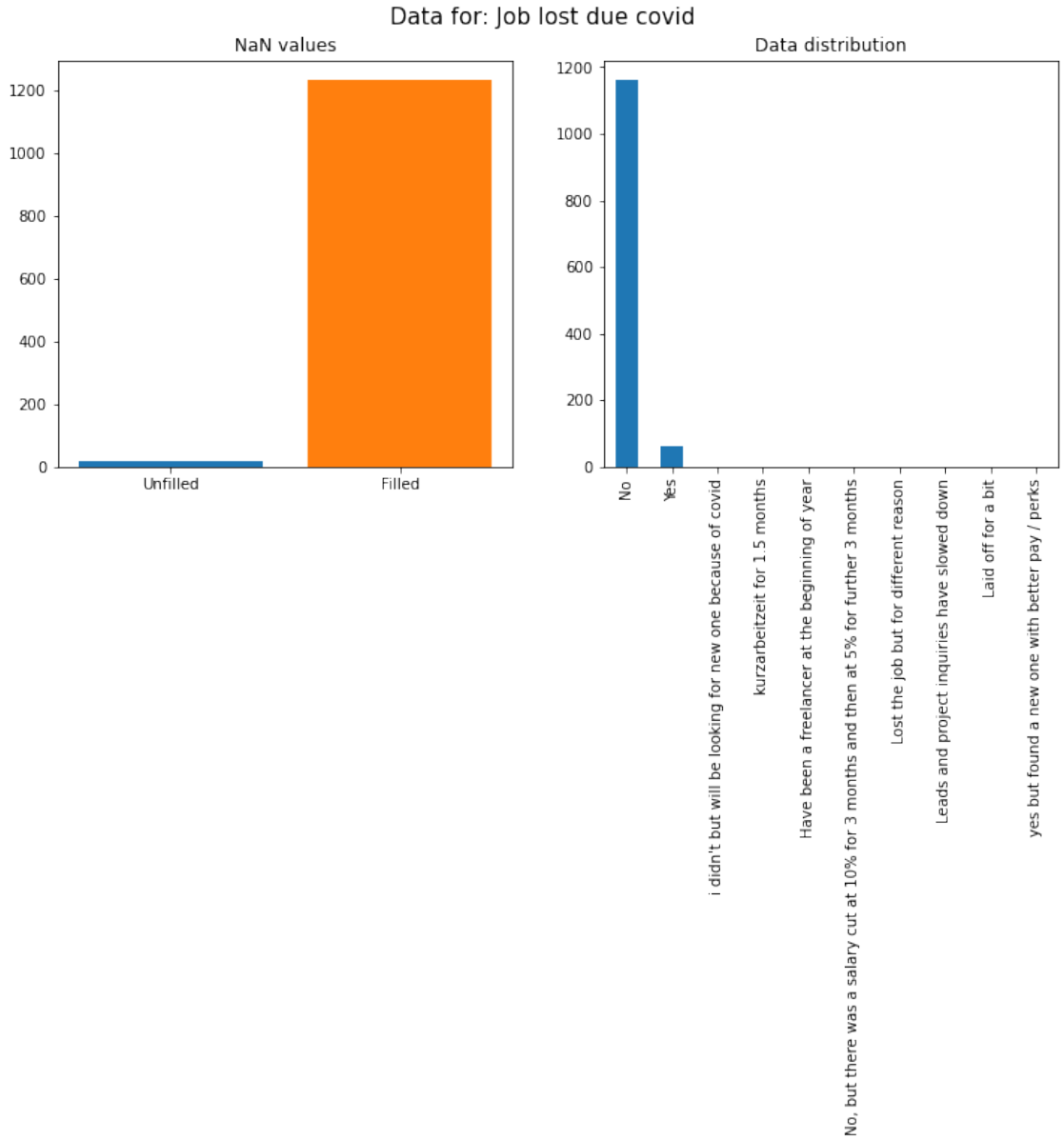
```
count      1235
unique        5
top       1000+
freq        448
Name: Company size, dtype: object
```



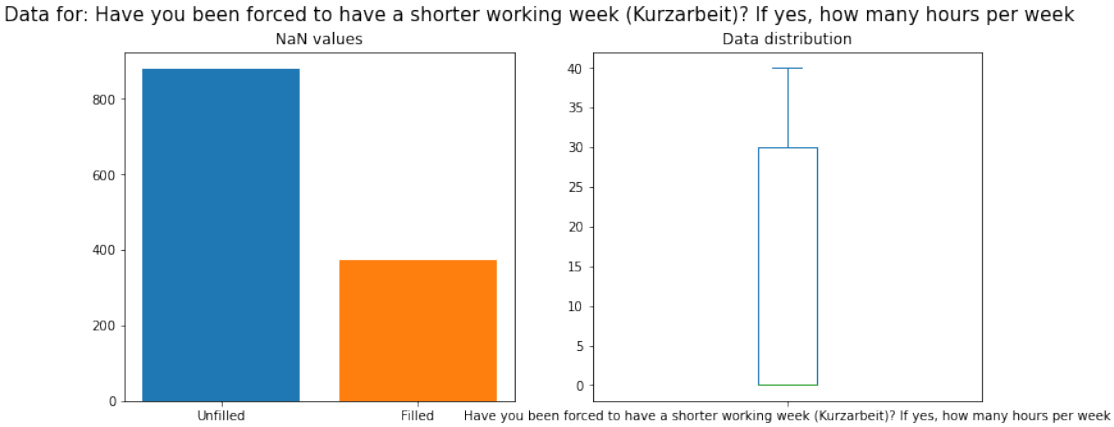

```
count        1228
unique         63
top       Product
freq          760
Name: Company type, dtype: object
```



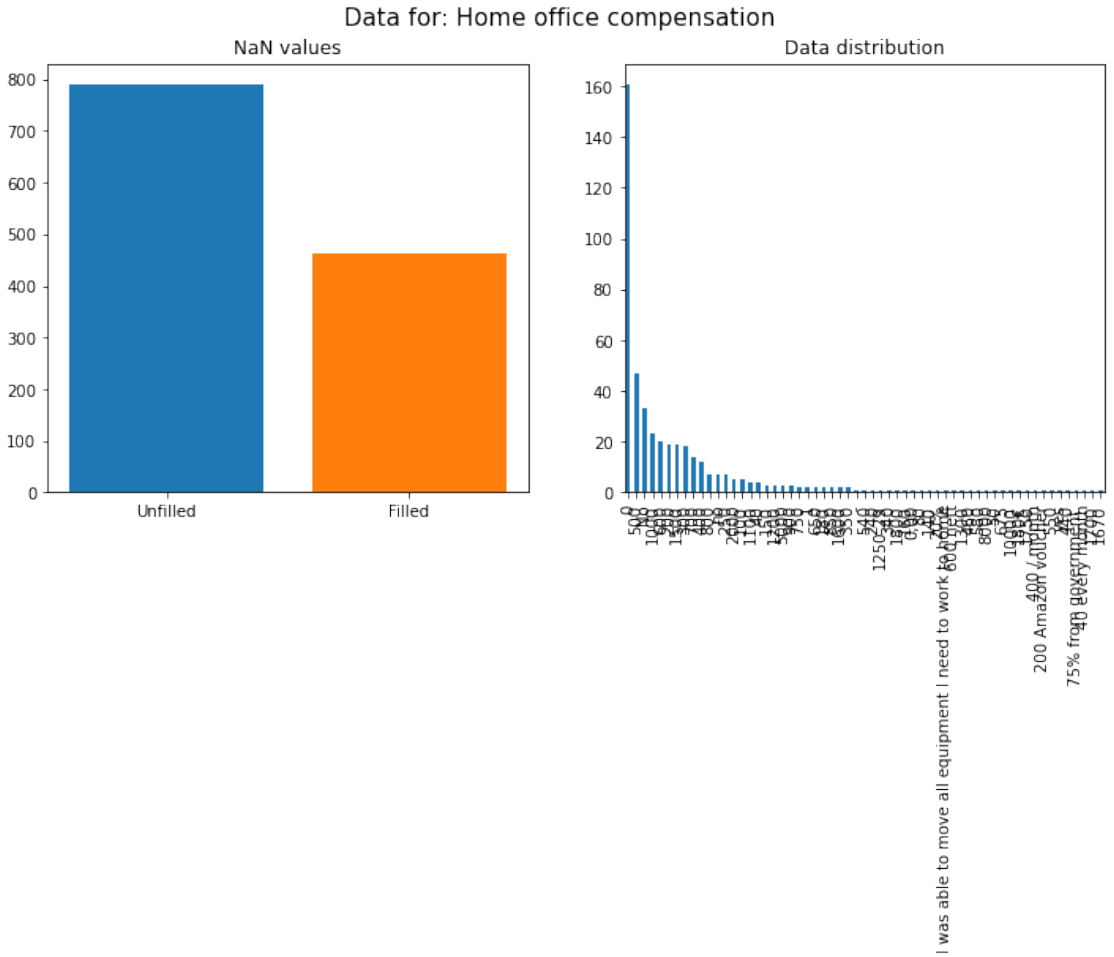

```
count     1233
unique      10
top         No
freq      1162
Name: Job lost due covid, dtype: object
```




```
count    373.000000
mean      12.967828
std       15.275174
min        0.000000
25%        0.000000
50%        0.000000
75%       30.000000
max       40.000000
Name: Have you been forced to have a shorter working week (Kurzarbeit)? If yes,
how many hours per week, dtype: float64
```




```
count      462
unique      59
top          0
freq       161
Name: Home office compensation, dtype: object
```



Z vygenerovaných grafů je vidět, že většina dat je vyplněná. V případě, že chybí v jistém sloupci mnoho dat tak se jedná většinou o věci jako bonusy, různé kompenzace a podobně. U těchto atributů se dá předpokládat, že když například uživatel nedostal kompenzaci za práci z domu tak položku nevyplnil. Pro další práci s datovou sadou si podrobně projdeme datové sloupce a zjistíme jaké konkrétní chyby se vyskytují v datové sadě, aby jsme se na ně mohli soustředit v další části čištění dat.

Age Většina hodnot je vyplněná. Dokonce jsou i v rozumném rozpětí, kde většina respondentů je ve věku 30 až 35 let. Jedná se o numerický atribut.

Gender Pohlaví také většina respondentů vyplnila a je rozděleno do tří kategorií, zde nebude nutné nějaké významné čištění, jen doplníme chybějící hodnoty nejčastější hodnotou. Jelikož se jedná o datovou sadu z technického oboru, tak není překvapující, že rozložení pohlaví je výrazně nevyvážené.

City Tento kategorický atribut bude potřeba pro další práci nějak shluknout. Podle hodnot je zřejmé, že sběr dat pochází z Německa.

Position Tento kategorický atribut bude také pro další práci potřeba vyčistit, konkrétně odfiltrovat hodnoty s nízkou četností.

Total years of experience Tento zřejmě numerický atribut byl knihovnou pandas interpretován jako kategorický z důvodu pár textových odpovědí. Pro naši další práci bude potřeba převést na numerický a textové odpovědi vyfiltrovat.

Years of experience in Germany Stejný případ jako u Total years of experience.

Seniority level Kategorický atribut, který se zjevně rozpadá do čtyř kategorií a dále mnoho odlehlých hodnot, které můžeme nejspíše shluknout.

Main Technology Pro nás se jedná o velmi významný sloupec, bohužel je velmi nešťastně zadaný. Pro naši další práci rozdělíme řetězce na jednotlivé technologie a zobrazíme si jejich histogram, z kterého vyčteme nejvíce používané technologie.

Other technologies Obdobný problém jako u main technology jen zde se uživatelé rozepsali ještě více a zde se jedná v této formě již o úplně nepoužitelný atribut

Yearly brutto Numerický atribut, z kterého zatím bohužel nic nevyčteme jelikož obsahuje dost výraznou anomálii, které se před další prací budeme muset zbavit.

Yearly bonus + stocks in EUR Zjevně numerický atribut, který je interpretován jako kategorický. Bude nutné převést na numerický. Mnoho nezadaných hodnot, ale z podstaty věci se dá předpokládat, že když někdo nevyplní bonus tak žádný nemá.

Annual brutto salary one year ago. Roční plat před jedním rokem, opět graf zatížen odlehlou hodnotou, kterou bude potřeba vyfiltrovat.

Bonus and stocks in same country Obdobně jako u Yearly bonus + stocks in EUR.

Number of vacation days Bude nutné převést na numerický atribut, jelikož naše datová sada obsahuje většinu lidí co jsou zaměstnanci tak chybějící hodnoty budeme doplňovat střední hodnotou.

Employment status Většina respondentů jsou zaměstnanci na plný úvazek. Můžeme pouze očistit o málo frekventované kategorie, zde jich ale naštěstí není tolik.

Contract duration Většina smluv je na dobu neurčitou. Vydíme jednu odlehlou hodnotu 0, které by bylo vhodné se zbavit a pár chybějících hodnot, které si můžeme v tomhle sloupci dovolit nahradit nejčetnější hodnotou, jelikož jasně převládá a chybějících hodnot je málo.

Main language at work Kategorický atribut, kde převládají dvě hodnoty. Zbytek je možno rozdělit separátorem a případně oddělat nesmyslné hodnoty.

Company size Nezašumělý kategorický atribut, kde pár chybějících hodnot můžeme nahradit nejčetnější hodnotou.

Company type Mnoho hodnot v tomto kategorickém atributu má malý výskyt. Dominují zde tři kategorie a zbytek bude vhodné shluknout.

Job lost due covid Zjevně binární atribut, který obsahuje mnoho slovních odpovědí, bude nutné očistit.

Have you been forced to have a shorter working week Mnoho chybějících hodnot, šlo by z tohoto vyčíst průměrnou dobu v práci, proto chybějící hodnoty nahradíme 40ti hodinami jako standardní pracovní týden, což je asi výchozí hodnota a člověk který nemá zkrácený pracovní úvazek tuto hodnotu zřejmě ignoroval.

Home office compensation Další zjevně numerický atribut, který bude potřeba očistit od slovních hodnot.

1.3.2 Korelační analýza

korelační analýza slouží pro hledání podobností mezi atributy. Když najdeme silnější korelaci mezi atributy, tak můžeme využít regresi pro dopočítání chybějících hodnot mezi takto korelovanými atributy a daný odhad bude daleko přesnější než například medián ze souboru. Korelační analýza pracuje nad numerickými atributy. Jak jsme si ukázali při analýze datových atributů tak mnoho numerických je interpretováno jako kategorické atributy z důvodu například nějaké textové odpovědi. Takto detekované atributy si pomocí knihovny pandas převedeme na numerické, kde špatné hodnoty převedeme na NaN.

[5]:

```
#
data['Total years of experience'] = pd.to_numeric(data['Total years of␣
 ↪experience'], errors="coerce")
data['Age'] = pd.to_numeric(data['Age'], errors="coerce")
data['Years of experience in Germany'] = pd.to_numeric(data['Years of␣
 ↪experience in Germany'], errors="coerce")
data['Yearly bonus + stocks in EUR'] = pd.to_numeric(data['Yearly bonus +␣
 ↪stocks in EUR'], errors="coerce")
data['Bonus and stocks in same country'] = pd.to_numeric(data['Bonus and stocks␣
 ↪in same country'], errors="coerce")
data['Number of vacation days'] = pd.to_numeric(data['Number of vacation␣
 ↪days'], errors="coerce")
data['Home office compensation'] = pd.to_numeric(data['Home office␣
 ↪compensation'], errors="coerce")

data.sample(5)
```

[5]:

	Age	Gender	City	Position	Total years of experience	\
259	35.0	Male	Frankfurt	Banker	12.0	
397	36.0	Male	Hamburg	Backend Developer	10.0	
441	32.0	Male	Berlin	CTO	10.0	
760	29.0	Female	Berlin	Software Engineer	5.0	
797	30.0	Male	Berlin	QA Engineer	8.0	

	Years of experience in Germany	Seniority level	Main Technology	\
259	2.0	Senior	NaN	
397	5.0	Senior	Java/Scala	
441	4.0	CTO	JAVA	
760	2.0	Middle	PHP	
797	3.0	Senior	Python	

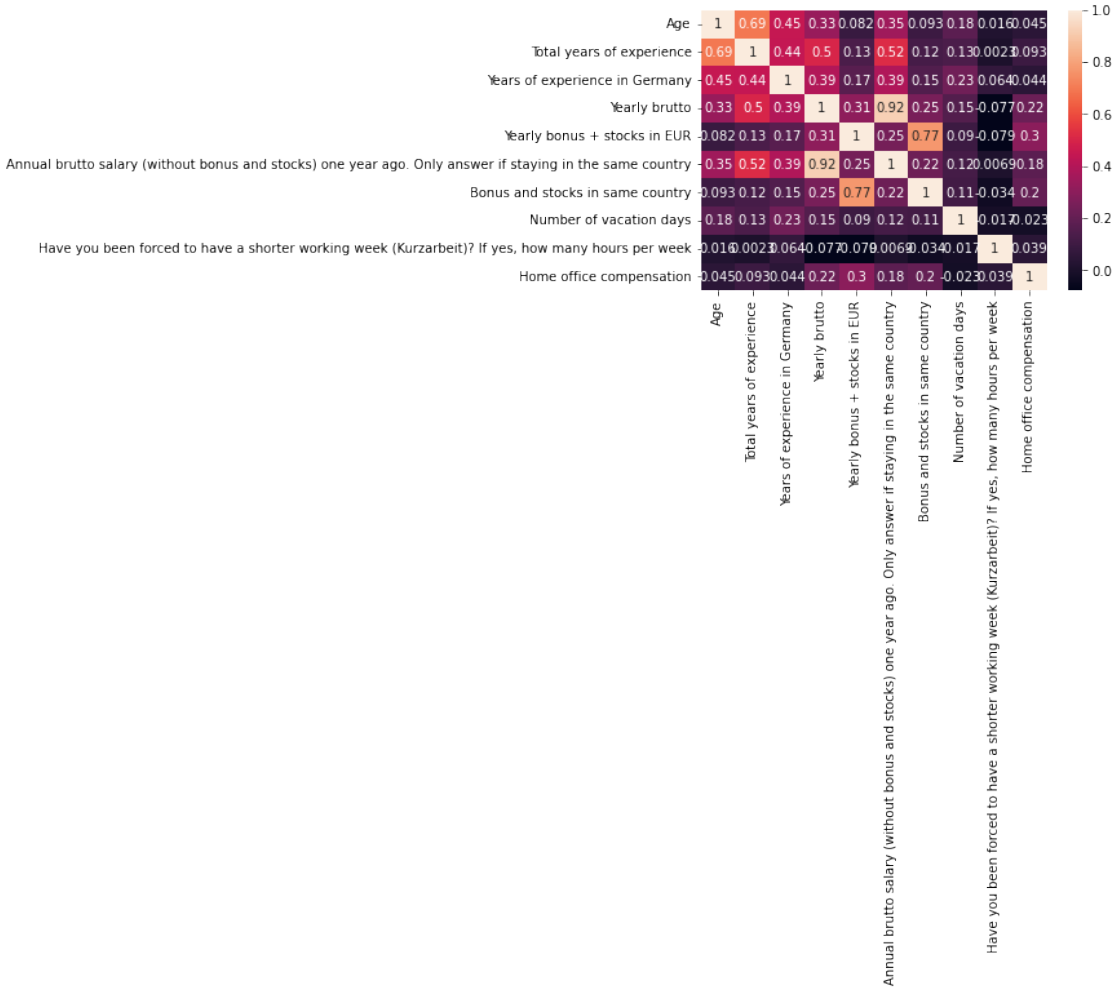
	Other technologies	Yearly brutto	\
259	NaN	100000.0	
397	Python, Go, AWS, Docker	82000.0	
441	Python, C/C++	200000.0	
760	Python, Javascript / Typescript, SQL, AWS, Goo…	55000.0	
797	Python, C/C++, Docker	68000.0	

	Yearly bonus + stocks in EUR	\
259	32000.0	
397	NaN	
441	200000.0	
760	NaN	
797	NaN	

	Annual brutto salary (without bonus and stocks) one year ago. Only answer if staying in the same country	\

```
259                                                 NaN
397                                             78000.0
441                                                 NaN
760                                             47000.0
797                                             66000.0

     Bonus and stocks in same country  Number of vacation days  \
259                               NaN                      NaN
397                               NaN                     30.0
441                               NaN                     28.0
760                               NaN                     24.0
797                               NaN                     30.0

      Employment status    ontract duration Main language at work  \
259  Full-time employee  Unlimited contract               English
397  Full-time employee  Unlimited contract                German
441  Full-time employee  Unlimited contract               English
760  Full-time employee  Unlimited contract               English
797  Full-time employee  Unlimited contract               English

    Company size Company type Job lost due covid  \
259     101-1000         Bank                 No
397       51-100      Product                 No
441     101-1000      Product                Yes
760        11-50      Startup                 No
797       51-100      Startup                 No

     Have you been forced to have a shorter working week (Kurzarbeit)? If yes,
how many hours per week  \
259                                                NaN
397                                                NaN
441                                                NaN
760                                                NaN
797                                                NaN

     Home office compensation
259                       NaN
397                     500.0
441                       NaN
760                       NaN
797                       NaN
```

[6]:
```
cov_matrix = data.corr('spearman')#pd.DataFrame.corr(data)
#print(cov_matrix)
sn.heatmap(cov_matrix, annot=True)
plt.show()
```



Z heat mapy můžeme vidět závislosti mezi věkem a odpracovanými roky což je pochopitelné a tuto závislost tedy budeme moci využít. Další silná závislost je mezi sloupci ohledně investic, které takto dopočítávat ale nebudeme.

1.4 Přípravy datové sady

Máme za sebou explorativní analýzu, v které jsme identifikovali jisté nedostatky v datové sadě. Nyní se pokusíme data očistit a doplnit. ### Ořezání prázdných záznamů a odfiltrování neužitečných atributů Zde odstraníme neužitečné atributy a zahodíme záznamy, které obsahují míň jak šest vyplněných hodnot, jelikož takové záznamy mají malou přídanou hodnotu pro datovou sadu a mnoho dat by bylo bráno jako průměr či nějak korelováno.

[7]:
```
data2 = data
if "Timestamp" in data.columns:
    data2 = data.drop('Timestamp', axis=1)

if 'Are you getting any Stock Options?' in data2.columns:
```

```
    data2 = data2.drop('Are you getting any Stock Options?', axis=1)


print("Records: ",len(data2))
data2 = data2.dropna(thresh=6)

print("Records: ",len(data2))
```

```
Records:  1253
Records:  1249
```

1.4.1 Dopočítání chybějících hodnot

V této části se pokusíme doplnit do datové sady chybějící hodnoty pomocí různých přístupů. Kde půjde odhadnout hodnotu na základě vysoké korelace s jiným atributem tak využijeme regresi. U sloupců, kde to dává smysl tak použijeme střední hodnotu. Jsou sloupce kde můžeme očekávat, že není vyplněno jelikož se to daného respondenta netýkalo (například chybějící položka bonusy ve firmě, tak zřejmě žádné bonusy nemá).

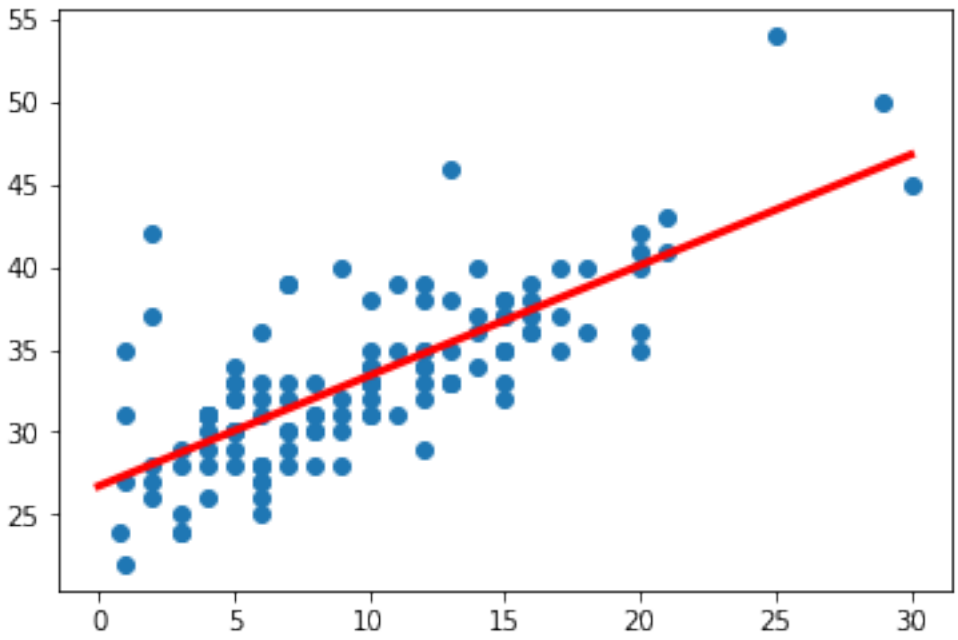
Tvorba regresních prediktorů Zde vytvoříme prediktory pro korelované atributy. Jako trénovací data vezmeme všechny záznamy bez chybějících hodnot.

[8]:
```
#convert object types to float, errors coerce specified if its not a numeric␣
 ↪type replace with NaN
data2['Total years of experience'] = pd.to_numeric(data2['Total years of␣
 ↪experience'], errors="coerce")
data2['Age'] = pd.to_numeric(data2['Age'], errors="coerce")

dataLearn = data2.dropna()

AgePredict = LinearRegression()
AgePredict = AgePredict.fit(dataLearn[['Total years of experience']].values,␣
 ↪dataLearn[['Age']].values)

YearOfExperiencePredict = LinearRegression()
YearOfExperiencePredict = YearOfExperiencePredict.fit(dataLearn[['Age']].
 ↪values, dataLearn[['Total years of experience']].values)


YearOfExperienceInGermanyPredict = LinearRegression()
YearOfExperienceInGermanyPredict = YearOfExperienceInGermanyPredict.
 ↪fit(dataLearn[['Total years of experience']].values, dataLearn[['Years of␣
 ↪experience in Germany']].values)
```

```
YearScore = YearOfExperiencePredict.score(dataLearn[['Age']].values,␣
 ↪dataLearn[['Total years of experience']].values)
print(YearScore)

x = np.linspace(0, 30, 30)
plt.plot(dataLearn['Total years of experience'], dataLearn['Age'], 'o')

AgeScore = AgePredict.score(dataLearn[['Total years of experience']].values,␣
 ↪dataLearn[['Age']].values)
print(AgeScore)

diabetes_y_pred = AgePredict.predict(np.array([x]).T)


plt.plot(x, diabetes_y_pred, color="red", linewidth=3)
```

```
0.5715946738931869
0.571594673893187
```

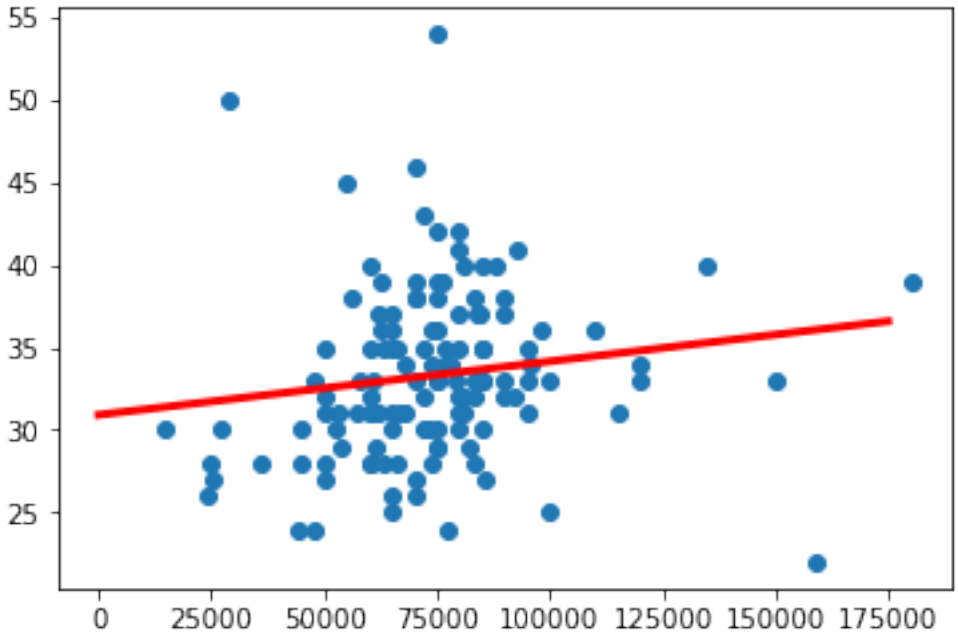
[8]: `[<matplotlib.lines.Line2D at 0x2b405cf85b0>]`



[9]:
```
salaryColumnName = ""

if "Current Salary" in dataLearn:
    salaryColumnName = "Current Salary"
elif "Yearly brutto" in dataLearn:
```

```
    salaryColumnName = "Yearly brutto"

AgePredictDependsOnSalary = LinearRegression()
AgePredictDependsOnSalary = AgePredictDependsOnSalary.
 ↪fit(dataLearn[[salaryColumnName]].values, dataLearn[['Age']].values)

x = np.linspace(0, 175000, 175000)
plt.plot(dataLearn[salaryColumnName], dataLearn['Age'], 'o')

AgePredictDependsOnSalary.score(dataLearn[[salaryColumnName]].values,␣
 ↪dataLearn[['Age']].values)


diabetes_y_pred = AgePredictDependsOnSalary.predict(np.array([x]).T)
plt.plot(x, diabetes_y_pred, color="red", linewidth=3)
```

[9]: [<matplotlib.lines.Line2D at 0x2b406651820>]



Tento prediktor se snažil naučit vztah mezi věkem a platem. Jak vidíme tak zde nějak silný vztah neexistuje. Použití takového prediktoru by vedlo k chybám.

1.4.2 Doplnění nekorelovaných atributů

[10]:
```
data2.loc[data2['Yearly bonus + stocks in EUR'].isna(), 'Yearly bonus + stocks␣
 ↪in EUR'] = 0
data2.loc[data2['Bonus and stocks in same country'].isna(), 'Bonus and stocks␣
 ↪in same country'] = 0
data2.loc[data2['Number of vacation days'].isna(), 'Number of vacation days'] =␣
 ↪data2['Number of vacation days'].mode()[0]
data2.loc[data2['Job lost due covid'].isna(), 'Job lost due covid'] = "No"
data2.loc[data2['Home office compensation'].isna(), 'Home office compensation']␣
 ↪= 0
data2.loc[data2['Have you been forced to have a shorter working week␣
 ↪(Kurzarbeit)? If yes, how many hours per week'].isna(), 'Have you been␣
 ↪forced to have a shorter working week (Kurzarbeit)? If yes, how many hours␣
 ↪per week'] = 40 #standard hours per week
data2.loc[data2['Have you been forced to have a shorter working week␣
 ↪(Kurzarbeit)? If yes, how many hours per week'] == 0, 'Have you been forced␣
 ↪to have a shorter working week (Kurzarbeit)? If yes, how many hours per␣
 ↪week'] = 40 #standard hours per week
data2.loc[data2['Gender'].isna(), 'Gender'] = data2['Gender'].mode()[0]
data2.loc[data2['City'].isna(), 'City'] = data2['City'].mode()[0]
data2.loc[data2['Seniority level'].isna(), 'Seniority level'] =␣
 ↪data2['Seniority level'].mode()[0]
data2.loc[data2['Main language at work'].isna(), 'Main language at work'] =␣
 ↪data2['Main language at work'].mode()[0]
data2.loc[data2['Company size'].isna(), 'Company size'] = data2['Company size'].
 ↪mode()[0]
data2.loc[data2['Company type'].isna(), 'Company type'] = data2['Company type'].
 ↪mode()[0]



data2.loc[data2['Employment status'].isna(), 'Employment status'] =␣
 ↪data2['Employment status'].mode()[0]
data2.loc[data2[' ontract duration'].isna(), ' ontract duration'] =␣
 ↪data2[' ontract duration'].mode()[0]

data2.drop(data2.loc[data2["Main Technology"].isna()].index, inplace=True)
data2.drop(data2.loc[data2["Other technologies"].isna()].index, inplace=True)
```

1.4.3 Doplnění korelovaných atributů pomocí prediktorů

[11]:
```
data2.loc[data2['Age'].isna() & data2['Total years of experience'].notna(),␣
 ↪'Age'] = \
data2.loc[data2['Age'].isna() & data2['Total years of experience'].notna(),␣
 ↪'Total years of experience'].apply(lambda exp : np.round(AgePredict.
 ↪predict([[exp]]))[0][0])
```

```
data2 = data2.drop(data2.loc[data2['Age'].isna() & data2['Total years of␣
 ↪experience'].isna()].index)
```

[12]:
```
data2.loc[data2['Years of experience in Germany'].isna() & data2['Total years␣
 ↪of experience'].notna(), 'Years of experience in Germany'] = \
data2.loc[data2['Years of experience in Germany'].isna() & data2['Total years␣
 ↪of experience'].notna(), 'Total years of experience'].apply(lambda exp : np.
 ↪round(YearOfExperienceInGermanyPredict.predict([[exp]]))[0][0])

data2 = data2.drop(data2.loc[data2['Years of experience in Germany'].isna() &␣
 ↪data2['Total years of experience'].isna()].index)
```

agregace pozice Zde se snažíme snížit počet různých kategorií pro pozici

[13]:
```
originalLenght = len(data2['Position'].unique())

for i in data2.index:
    if not pd.isnull(data2['Position'][i]):
        data2.loc[i, 'Position'] = re.
 ↪sub("(\s*senior\s*|\s*junior\s*|\s*middle\s*)", " ", data2['Position'][i].
 ↪lower())
        data2.loc[i, 'Position'] = re.sub("(^\s*|\s*$)", " ",␣
 ↪data2['Position'][i])
        data2.loc[i, 'Position'] = re.sub("\s+", " ", data2['Position'][i])

data2.drop(data2.loc[data2['Position'].isna()].index, inplace=True)
NewLenght = len(data2['Position'].unique())
print("Reduced", originalLenght-NewLenght)
```

```
Reduced 10
```

[14]:
```
data2.loc[data2['Total years of experience'].isna() & data2['Age'].notna(),␣
 ↪'Total years of experience'] = \
data2.loc[data2['Total years of experience'].isna() & data2['Age'].notna(),␣
 ↪'Age'].apply(lambda age : np.round(YearOfExperiencePredict.
 ↪predict([[age]]))[0][0])

data2.drop(data2.loc[data2['Total years of experience'].isna() & data2['Age'].
 ↪isna()].index, inplace=True)
```

Doplnění platu Zde nejspíše člověk neuvedl druhý plat, jelikož se mu plat nezměnil a přišlo mu to zbytečné. Můžeme křížově doplnit.

[15]:
```
originalLenght = len(data2)
```

```
currentSalary = "Yearly brutto"
salaryOneYearAgo = "Annual brutto salary (without bonus and stocks) one year␣
 ↪ago. Only answer if staying in the same country"


data2.loc[data2[currentSalary].isna() & data2[salaryOneYearAgo].notna(),␣
 ↪currentSalary] = data2.loc[data2[currentSalary].isna() &␣
 ↪data2[salaryOneYearAgo].notna(), salaryOneYearAgo]
data2.loc[data2[salaryOneYearAgo].isna() & data2[currentSalary].notna(),␣
 ↪salaryOneYearAgo] = data2.loc[data2[salaryOneYearAgo].isna() &␣
 ↪data2[currentSalary].notna(), currentSalary]

data2.drop(data2.loc[data2[currentSalary].isna() & data2[salaryOneYearAgo].
 ↪isna()].index, inplace=True)


"""



salaries = {
    'Current Salary': ['Salary one year ago', 'Salary two years ago'],
    'Salary one year ago': ['Salary two years ago', 'Current Salary'],
    'Salary two years ago': ['Salary one year ago', 'Current Salary']
}

for i in data2.index:
    for j in salaries.keys():
        if pd.isnull(data2[j][i]):
            for salary in salaries[j]:
                if not pd.isnull(data2[salary][i]):
                    data2.loc[i, j] = data2[salary][i]
                    break

    if pd.isnull(data2['Current Salary'][i]):
        data2.drop(i, inplace=True)

"""


NewLenght = len(data2)
print("Reduced", originalLenght-NewLenght)
```

```
Reduced 0
```

[16]:
```
data2.Position.str.split(expand=True).stack().value_counts()
'''
```

```
print(x.keys()[0], x[0])

dataLearn = data2.dropna()

AgePredict = LinearRegression()
AgePredict = AgePredict.fit(x.keys()[0], dataLearn[['Age']].values)

x = np.linspace(0, 30, 30)
plt.plot(dataLearn['Position'], dataLearn['Age'], 'o')

AgeScore = AgePredict.score(x.keys()[0], dataLearn[['Age']].values)
print(AgeScore)

diabetes_y_pred = AgePredict.predict(np.array([x]).T)
plt.plot(x, diabetes_y_pred, color="red", linewidth=3)
'''
```

[16]:
```
'\nprint(x.keys()[0], x[0])\n\ndataLearn = data2.dropna()\n\nAgePredict =
LinearRegression()\nAgePredict = AgePredict.fit(x.keys()[0],
dataLearn[[\'Age\']].values)\n\nx = np.linspace(0, 30,
30)\nplt.plot(dataLearn[\'Position\'], dataLearn[\'Age\'], \'o\')\n\nAgeScore =
AgePredict.score(x.keys()[0],
dataLearn[[\'Age\']].values)\nprint(AgeScore)\n\ndiabetes_y_pred =
AgePredict.predict(np.array([x]).T)\nplt.plot(x, diabetes_y_pred, color="red",
linewidth=3)\n'
```

1.4.4 Detekce anomalií

V explorativní analýze jsme narazili na některé nečitelné numerické grafy, jelikož měřítko na ose výrazně ovlivnila odlehlá hodnota. Nyní se takovýchto hodnot pokusíme zbavit pomocí metody zscore. ta předpokládá normální rozdělení. V našem okolí se většina jevů chová podle normálního rozdělení, což nám blíže říká centrální limitní věta. V normálním rozdělení 99.7% hodnot leží v intervalu <-3 ; 3 > a oproto hodnoty mimo tento interval odstraníme.

[17]:
```
originalLenght = len(data2)


#print("dtypes: ", data.dtypes)
print("columns: ", data2.select_dtypes(include=["number"]).columns)

def drop_numerical_outliers(df, z_thresh=3):
    # Constrains will contain `True` or `False` depending on if it is a value␣
 ↪below the threshold.
    constrains = df.select_dtypes(include=["number"]) \
        .apply(lambda x: np.abs(stats.zscore(x)) < z_thresh) \
        .all(axis=1)
    # Drop (inplace) values set to be rejected
```

```
    df.drop(df.index[~constrains], inplace=True)

drop_numerical_outliers(data2)



NewLenght = len(data2)
print("Reduced: ", originalLenght-NewLenght)
data2.head()
```

```
columns:  Index(['Age', 'Total years of experience', 'Years of experience in
Germany',
       'Yearly brutto', 'Yearly bonus + stocks in EUR',
       'Annual brutto salary (without bonus and stocks) one year ago. Only
answer if staying in the same country',
       'Bonus and stocks in same country', 'Number of vacation days',
       'Have you been forced to have a shorter working week (Kurzarbeit)? If
yes, how many hours per week',
       'Home office compensation'],
      dtype='object')
Reduced:  135
```

[17]:

	Age	Gender	City	Position	Total years of experience	\
0	26.0	Male	Munich	software engineer	5.0	
4	37.0	Male	Berlin	backend developer	17.0	
5	32.0	Male	Berlin	devops	5.0	
7	24.0	Male	Berlin	frontend developer	5.0	
8	29.0	Male	Berlin	backend developer	8.0	

	Years of experience in Germany	Seniority level	Main Technology	\
0	3.0	Senior	TypeScript	
4	6.0	Senior	C# .NET	
5	1.0	Senior	AWS, GCP, Python,K8s	
7	1.0	Senior	Typescript	
8	2.0	Senior	PHP	

	Other technologies	Yearly brutto	\
0	Kotlin, Javascript / Typescript	80000.0	
4	.NET, SQL, AWS, Docker	62000.0	
5	Python, AWS, Google Cloud, Kubernetes, Docker	76000.0	
7	Javascript / Typescript	65000.0	
8	SQL, AWS, Docker	56000.0	

	Yearly bonus + stocks in EUR	\
0	5000.0	
4	0.0	
5	5000.0	

```
7                          0.0
8                          0.0

   Annual brutto salary (without bonus and stocks) one year ago. Only answer if
staying in the same country  \
0                                          75000.0
4                                          62000.0
5                                          76000.0
7                                          65000.0
8                                          55000.0

   Bonus and stocks in same country  Number of vacation days  \
0                           10000.0                     30.0
4                               0.0                     29.0
5                            5000.0                     30.0
7                               0.0                     27.0
8                               0.0                     28.0

    Employment status    ontract duration Main language at work Company size  \
0  Full-time employee  Unlimited contract               English       51-100
4  Full-time employee  Unlimited contract               English     101-1000
5  Full-time employee  Unlimited contract               English        11-50
7  Full-time employee  Unlimited contract               English        1000+
8  Full-time employee  Unlimited contract               English     101-1000

  Company type Job lost due covid  \
0      Product                 No
4      Product                 No
5      Startup                 No
7      Product                 No
8      Product                 No

   Have you been forced to have a shorter working week (Kurzarbeit)? If yes, how
many hours per week  \
0                                           40.0
4                                           40.0
5                                           40.0
7                                           40.0
8                                           30.0

   Home office compensation
0                       0.0
4                       0.0
5                       0.0
7                     600.0
8                       0.0
```

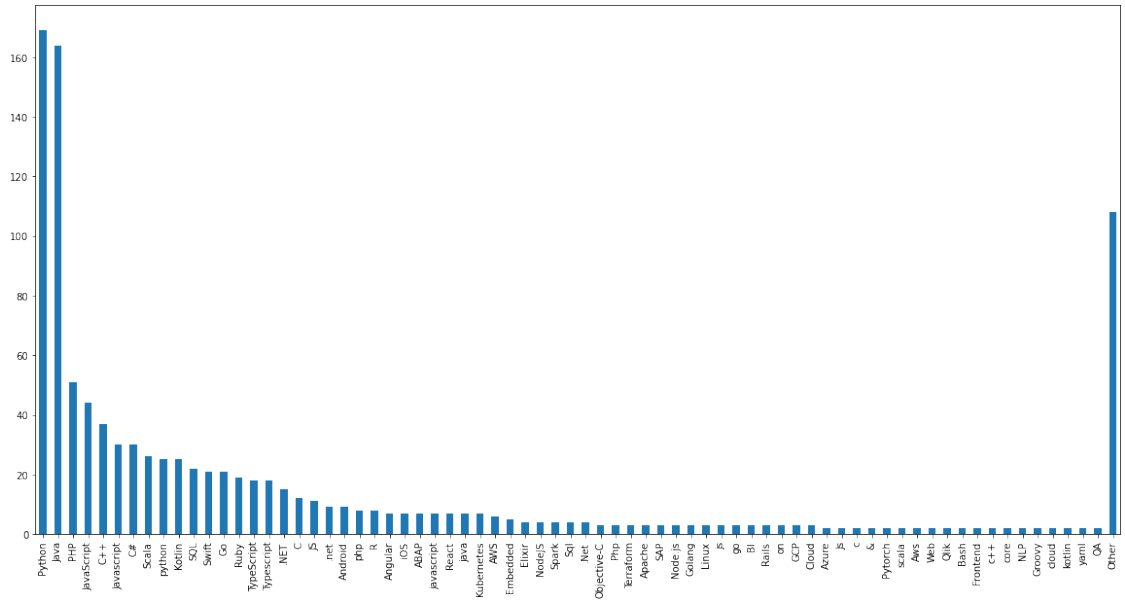
Výpis nejpoužívanějších technologií V explorativní analýze jsme detekovali, že mnoho lidí zadalo více technologií. Nyní se pokusíme hodnoty rozdělit a vykreslit graf četností použití technologií.

[18]:
```
#main technologies

mainTechs = data2["Main Technology"].str.split(pat=r"[,\/\s]+",expand=True).
 ↪stack().value_counts()

outliers = mainTechs.loc[mainTechs <= 1]
outliersCount = mainTechs.loc[mainTechs <= 1].sum()

del mainTechs['']

plt.rcParams["figure.figsize"] = (20,10)
mainTechs = mainTechs.drop(mainTechs.loc[mainTechs.isin(outliers)].index)
mainTechs["Other"] = outliersCount

mainTechs.plot(kind='bar')
plt.xticks(rotation=90)
```

[18]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
        51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67,
        68, 69, 70, 71]),
 [Text(0, 0, 'Python'),
  Text(1, 0, 'Java'),
  Text(2, 0, 'PHP'),
  Text(3, 0, 'JavaScript'),
  Text(4, 0, 'C++'),
  Text(5, 0, 'Javascript'),
  Text(6, 0, 'C#'),
  Text(7, 0, 'Scala'),
  Text(8, 0, 'python'),
  Text(9, 0, 'Kotlin'),
  Text(10, 0, 'SQL'),
  Text(11, 0, 'Swift'),
  Text(12, 0, 'Go'),
  Text(13, 0, 'Ruby'),
  Text(14, 0, 'TypeScript'),
  Text(15, 0, 'Typescript'),
  Text(16, 0, '.NET'),
  Text(17, 0, 'C'),
  Text(18, 0, 'JS'),
  Text(19, 0, '.net'),
  Text(20, 0, 'Android'),
```

```
Text(21, 0, 'php'),
Text(22, 0, 'R'),
Text(23, 0, 'Angular'),
Text(24, 0, 'iOS'),
Text(25, 0, 'ABAP'),
Text(26, 0, 'javascript'),
Text(27, 0, 'React'),
Text(28, 0, 'java'),
Text(29, 0, 'Kubernetes'),
Text(30, 0, 'AWS'),
Text(31, 0, 'Embedded'),
Text(32, 0, 'Elixir'),
Text(33, 0, 'NodeJS'),
Text(34, 0, 'Spark'),
Text(35, 0, 'Sql'),
Text(36, 0, '.Net'),
Text(37, 0, 'Objective-C'),
Text(38, 0, 'Php'),
Text(39, 0, 'Terraform'),
Text(40, 0, 'Apache'),
Text(41, 0, 'SAP'),
Text(42, 0, 'Node.js'),
Text(43, 0, 'Golang'),
Text(44, 0, 'Linux'),
Text(45, 0, 'js'),
Text(46, 0, 'go'),
Text(47, 0, 'BI'),
Text(48, 0, 'Rails'),
Text(49, 0, 'on'),
Text(50, 0, 'GCP'),
Text(51, 0, 'Cloud'),
Text(52, 0, 'Azure'),
Text(53, 0, 'Js'),
Text(54, 0, 'c'),
Text(55, 0, '&'),
Text(56, 0, 'Pytorch'),
Text(57, 0, 'scala'),
Text(58, 0, 'Aws'),
Text(59, 0, 'Web'),
Text(60, 0, 'Qlik'),
Text(61, 0, 'Bash'),
Text(62, 0, 'Frontend'),
Text(63, 0, 'c++'),
Text(64, 0, 'core'),
Text(65, 0, 'NLP'),
Text(66, 0, 'Groovy'),
Text(67, 0, 'cloud'),
```

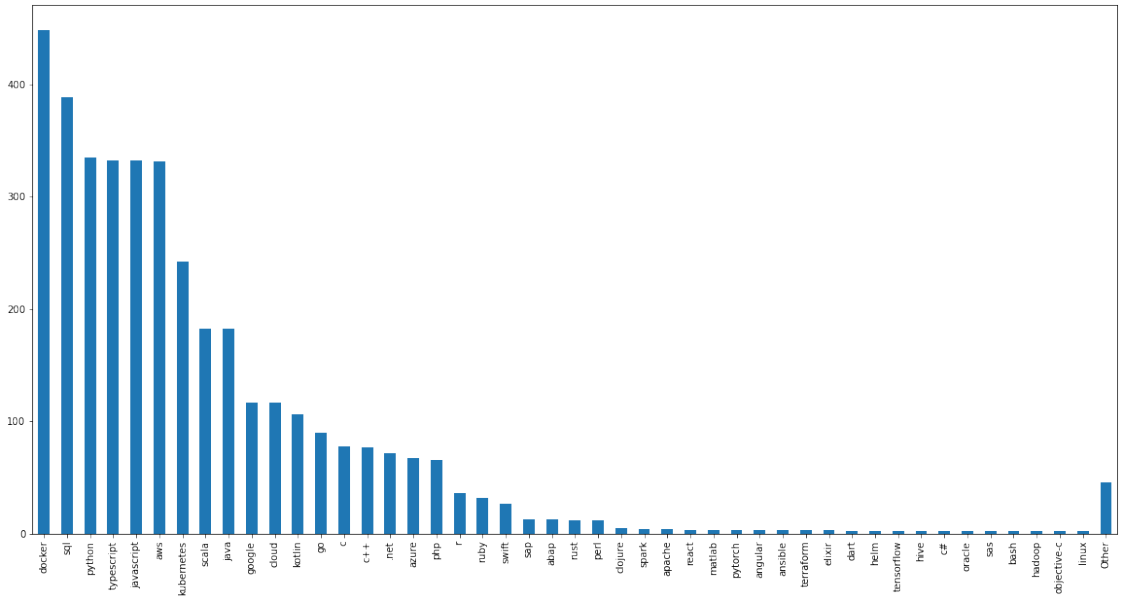
```
Text(68, 0, 'kotlin'),
Text(69, 0, 'yaml'),
Text(70, 0, 'QA'),
Text(71, 0, 'Other')])
```



[19]:
```
#other technologies

otherTechs = data2["Other technologies"].str.lower().str.split(pat=r"[,\/
 ↪\s]+",expand=True).stack().value_counts()

outliers = otherTechs.loc[otherTechs <= 1]
outliersCount = otherTechs.loc[otherTechs <= 1].sum()

del otherTechs['']

otherTechs = otherTechs.drop(otherTechs.loc[otherTechs.isin(outliers)].index)
otherTechs["Other"] = outliersCount

otherTechs.plot(kind='bar')
plt.xticks(rotation=90)
```

[19]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46]),
 [Text(0, 0, 'docker'),
  Text(1, 0, 'sql'),
  Text(2, 0, 'python'),
```

```
Text(3, 0, 'typescript'),
Text(4, 0, 'javascript'),
Text(5, 0, 'aws'),
Text(6, 0, 'kubernetes'),
Text(7, 0, 'scala'),
Text(8, 0, 'java'),
Text(9, 0, 'google'),
Text(10, 0, 'cloud'),
Text(11, 0, 'kotlin'),
Text(12, 0, 'go'),
Text(13, 0, 'c'),
Text(14, 0, 'c++'),
Text(15, 0, '.net'),
Text(16, 0, 'azure'),
Text(17, 0, 'php'),
Text(18, 0, 'r'),
Text(19, 0, 'ruby'),
Text(20, 0, 'swift'),
Text(21, 0, 'sap'),
Text(22, 0, 'abap'),
Text(23, 0, 'rust'),
Text(24, 0, 'perl'),
Text(25, 0, 'clojure'),
Text(26, 0, 'spark'),
Text(27, 0, 'apache'),
Text(28, 0, 'react'),
Text(29, 0, 'matlab'),
Text(30, 0, 'pytorch'),
Text(31, 0, 'angular'),
Text(32, 0, 'ansible'),
Text(33, 0, 'terraform'),
Text(34, 0, 'elixir'),
Text(35, 0, 'dart'),
Text(36, 0, 'helm'),
Text(37, 0, 'tensorflow'),
Text(38, 0, 'hive'),
Text(39, 0, 'c#'),
Text(40, 0, 'oracle'),
Text(41, 0, 'sas'),
Text(42, 0, 'bash'),
Text(43, 0, 'hadoop'),
Text(44, 0, 'objective-c'),
Text(45, 0, 'linux'),
Text(46, 0, 'Other')])
```

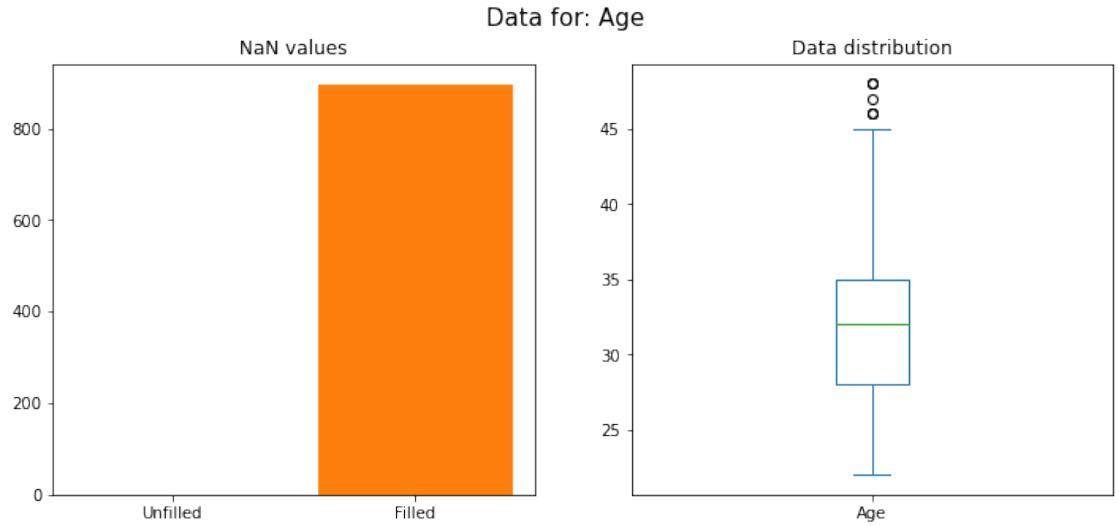


1.4.5 Kontrola po čištění

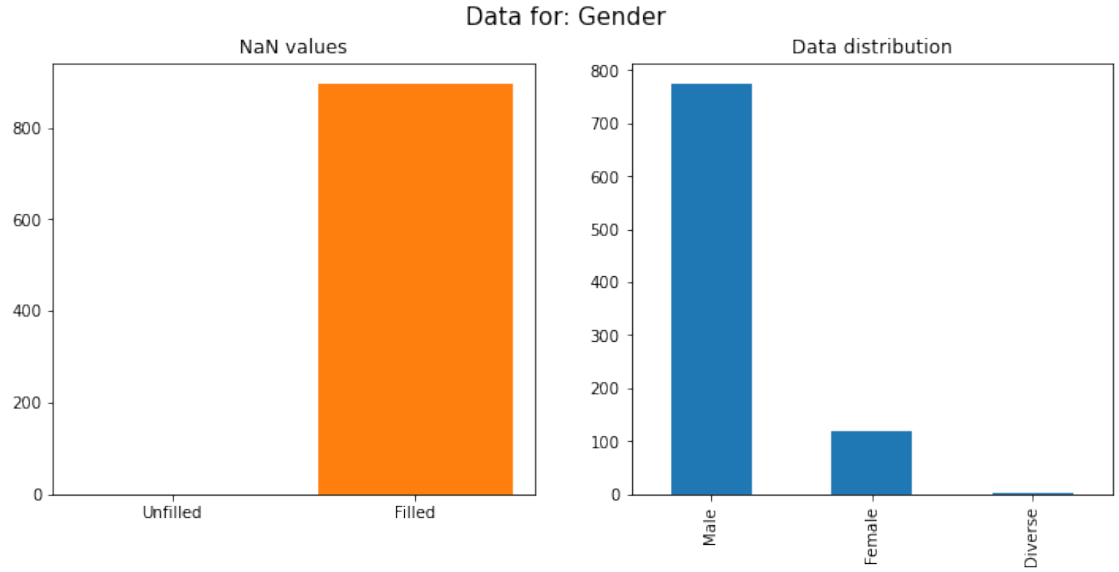
Nyní si zobrazíme očištěná data. Grafy by měli být čitelnější než v explorativní analýze.

[20]:
```
plot_graphs(data2)
```

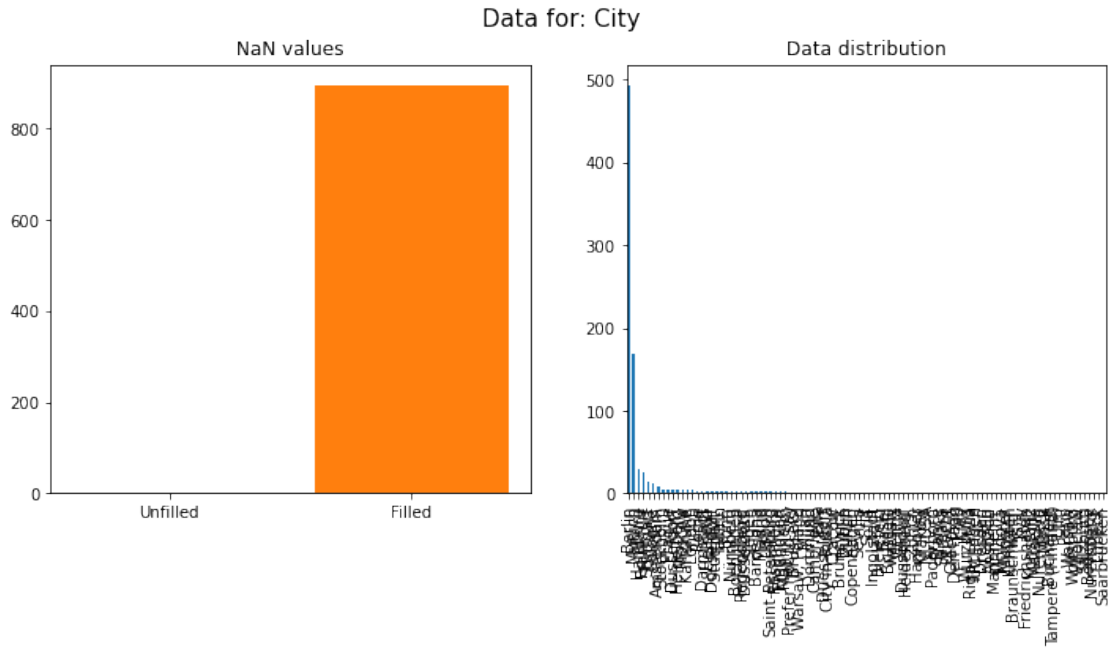
```
count    896.000000
mean      31.930804
std        4.808246
min       22.000000
25%       28.000000
50%       32.000000
75%       35.000000
max       48.000000
Name: Age, dtype: float64
```



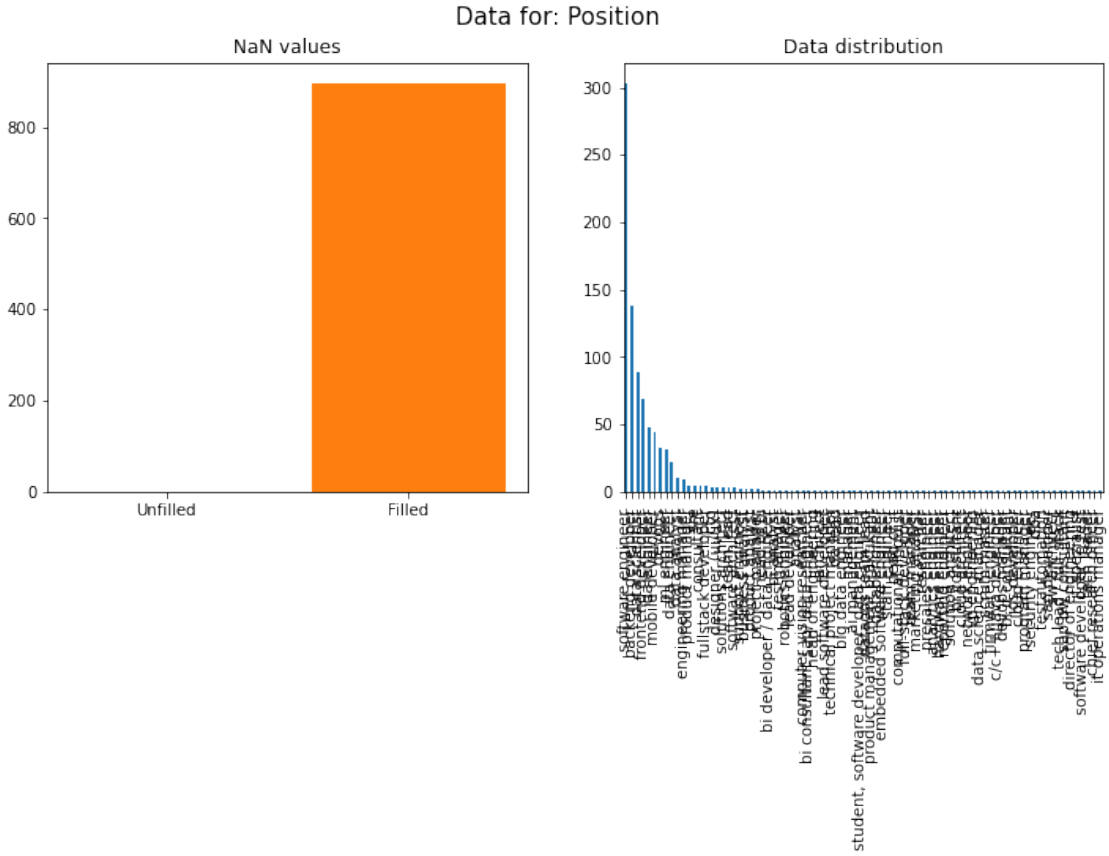

```
count      896
unique       3
top       Male
freq       775
Name: Gender, dtype: object
```



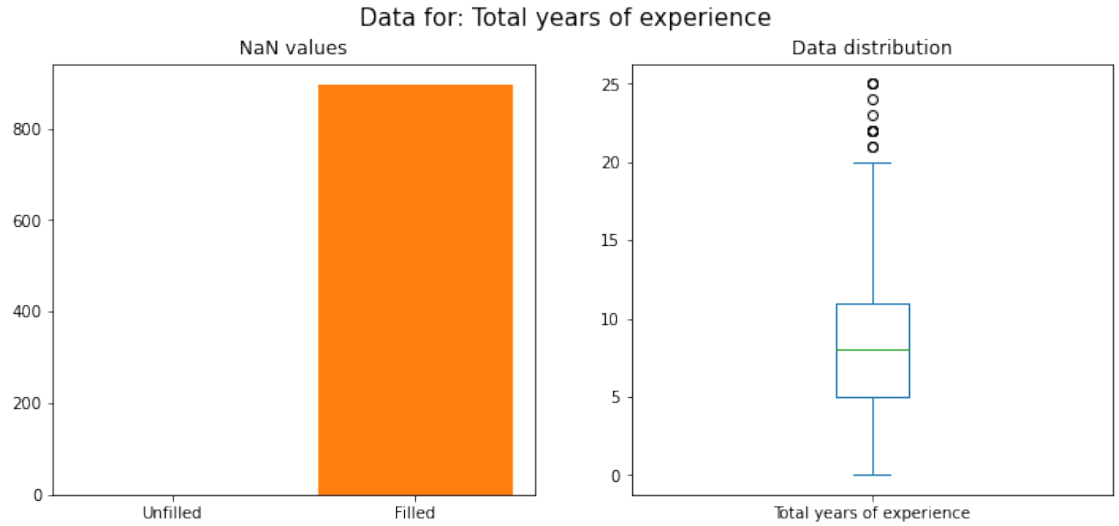

```
count        896
unique        98
top       Berlin
freq         494
Name: City, dtype: object
```



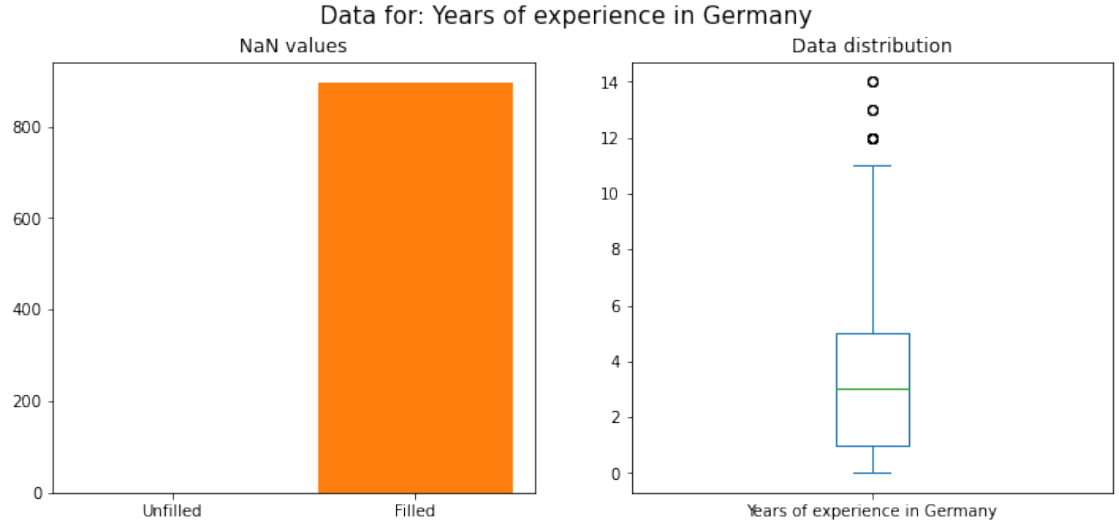

```
count                    896
unique                    84
top        software engineer
freq                     303
Name: Position, dtype: object
```




```
count    896.000000
mean       8.345536
std        4.737489
min        0.000000
25%        5.000000
50%        8.000000
75%       11.000000
max       25.000000
Name: Total years of experience, dtype: float64
```

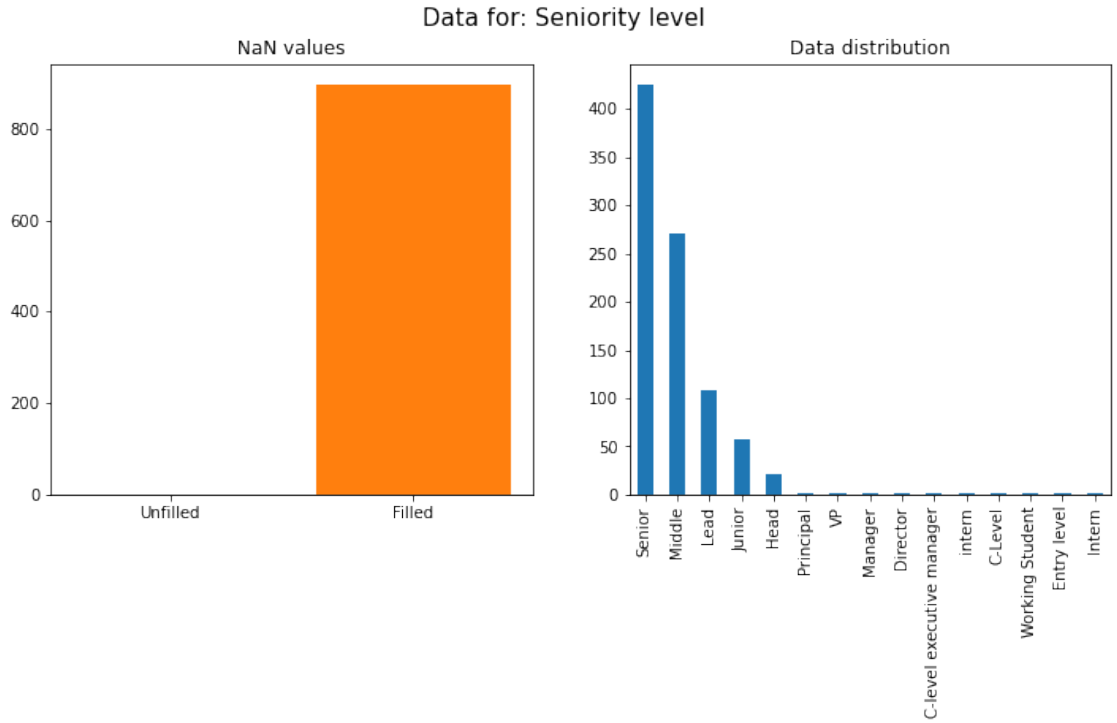



```
count    896.000000
mean       3.373996
std        2.691107
min        0.000000
25%        1.000000
50%        3.000000
75%        5.000000
max       14.000000
Name: Years of experience in Germany, dtype: float64
```

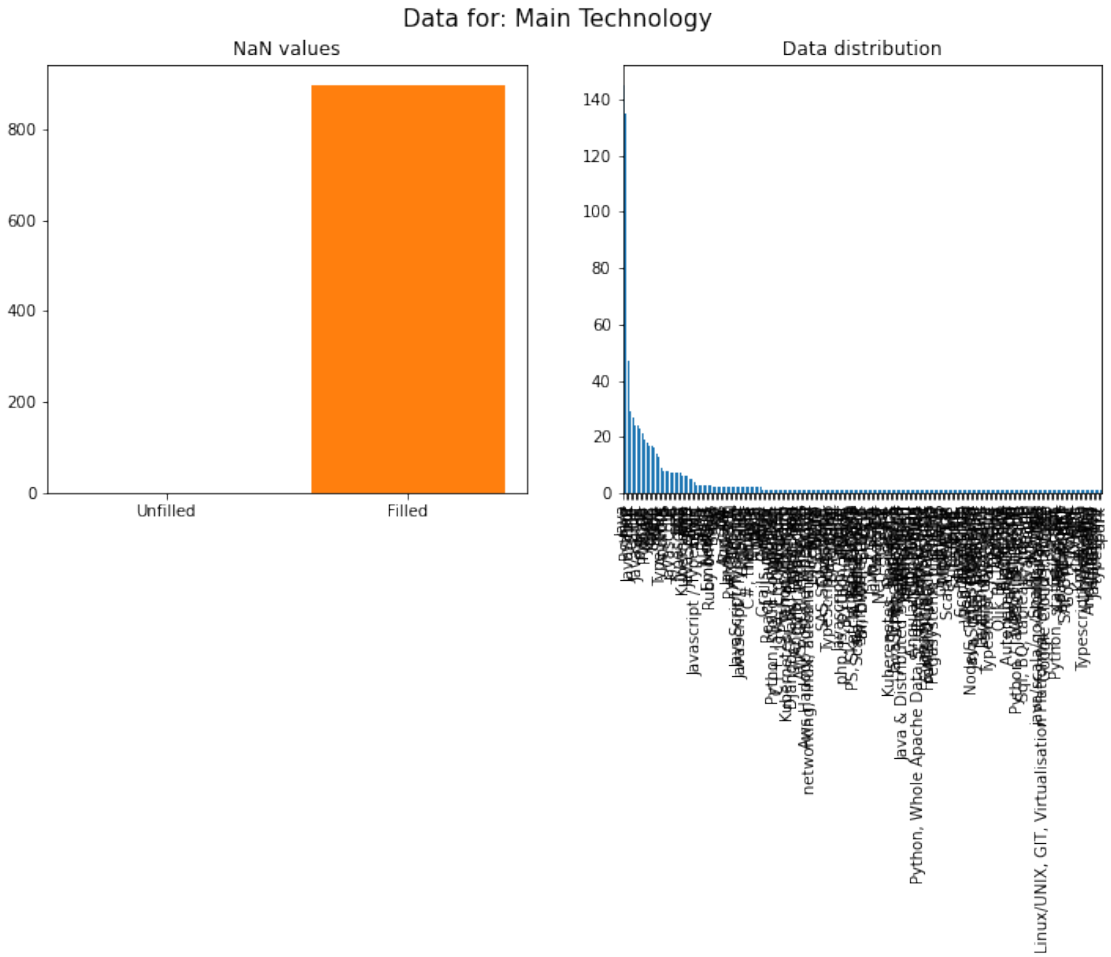



```
count        896
unique        15
```

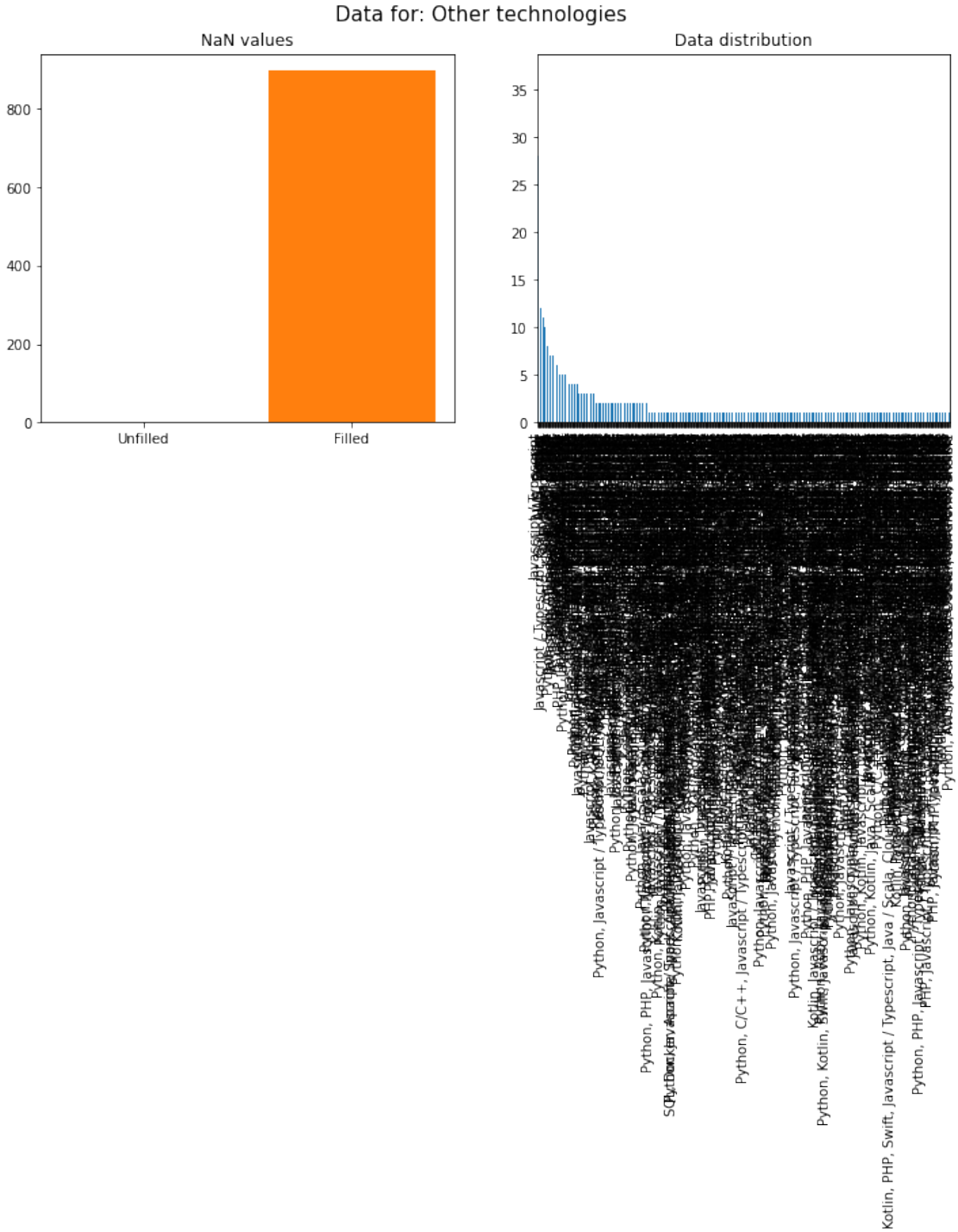
```
top       Senior
freq         425
Name: Seniority level, dtype: object
```




```
count      896
unique     203
top       Java
freq       145
Name: Main Technology, dtype: object
```

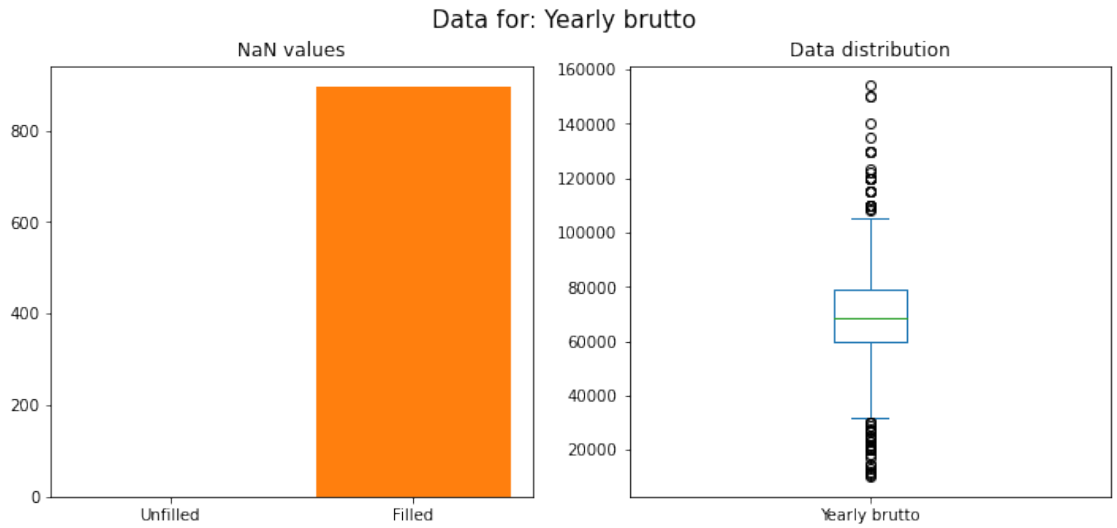



```
count                          896
unique                         475
top       Javascript / Typescript
freq                            37
Name: Other technologies, dtype: object
```

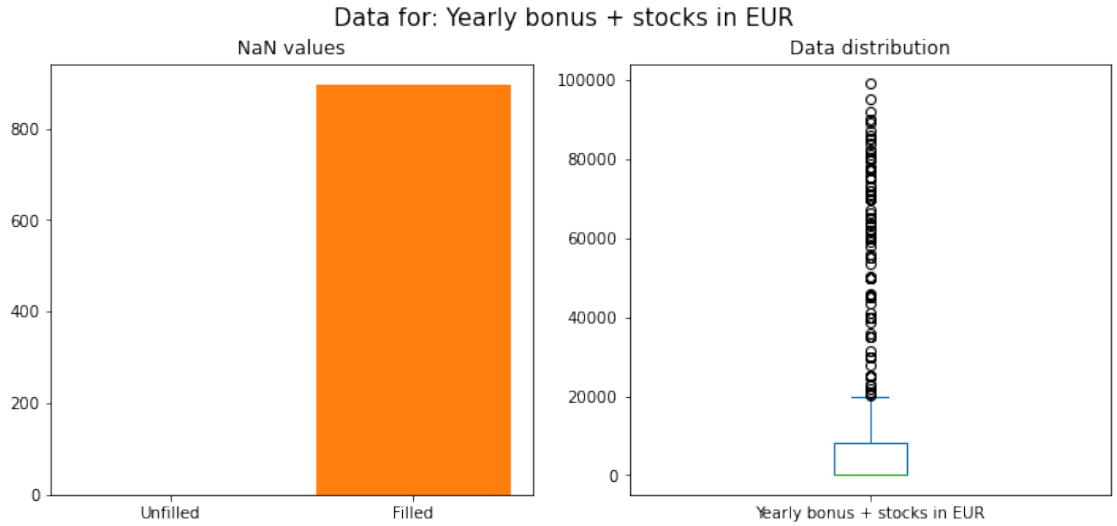



```
count       896.00000
mean      68957.67202
std       19088.11889
min       10001.00000
```

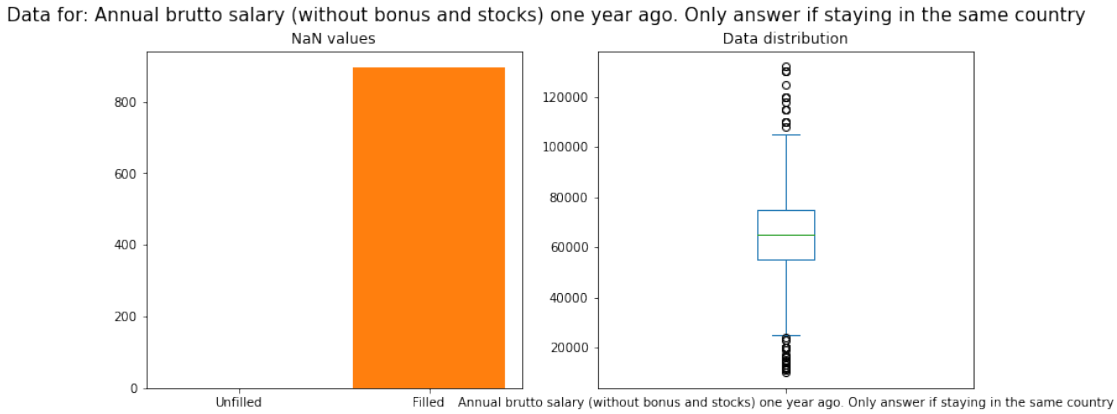
```
25%       60000.00000
50%       68500.00000
75%       79000.00000
max      154000.00000
Name: Yearly brutto, dtype: float64
```




```
count      896.000000
mean     10308.708092
std      21015.467597
min          0.000000
25%          0.000000
50%          0.000000
75%       8000.000000
max      99000.000000
Name: Yearly bonus + stocks in EUR, dtype: float64
```



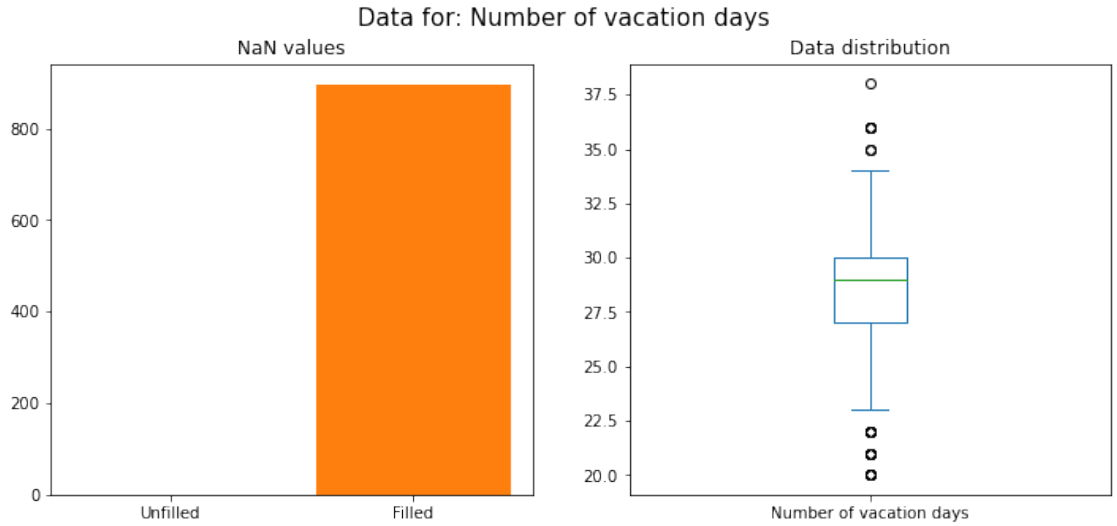

```
count       896.000000
mean      64206.965547
std       17717.972456
min       10001.000000
25%       55000.000000
50%       65000.000000
75%       75000.000000
max      132000.000000
Name: Annual brutto salary (without bonus and stocks) one year ago. Only answer
if staying in the same country, dtype: float64
```



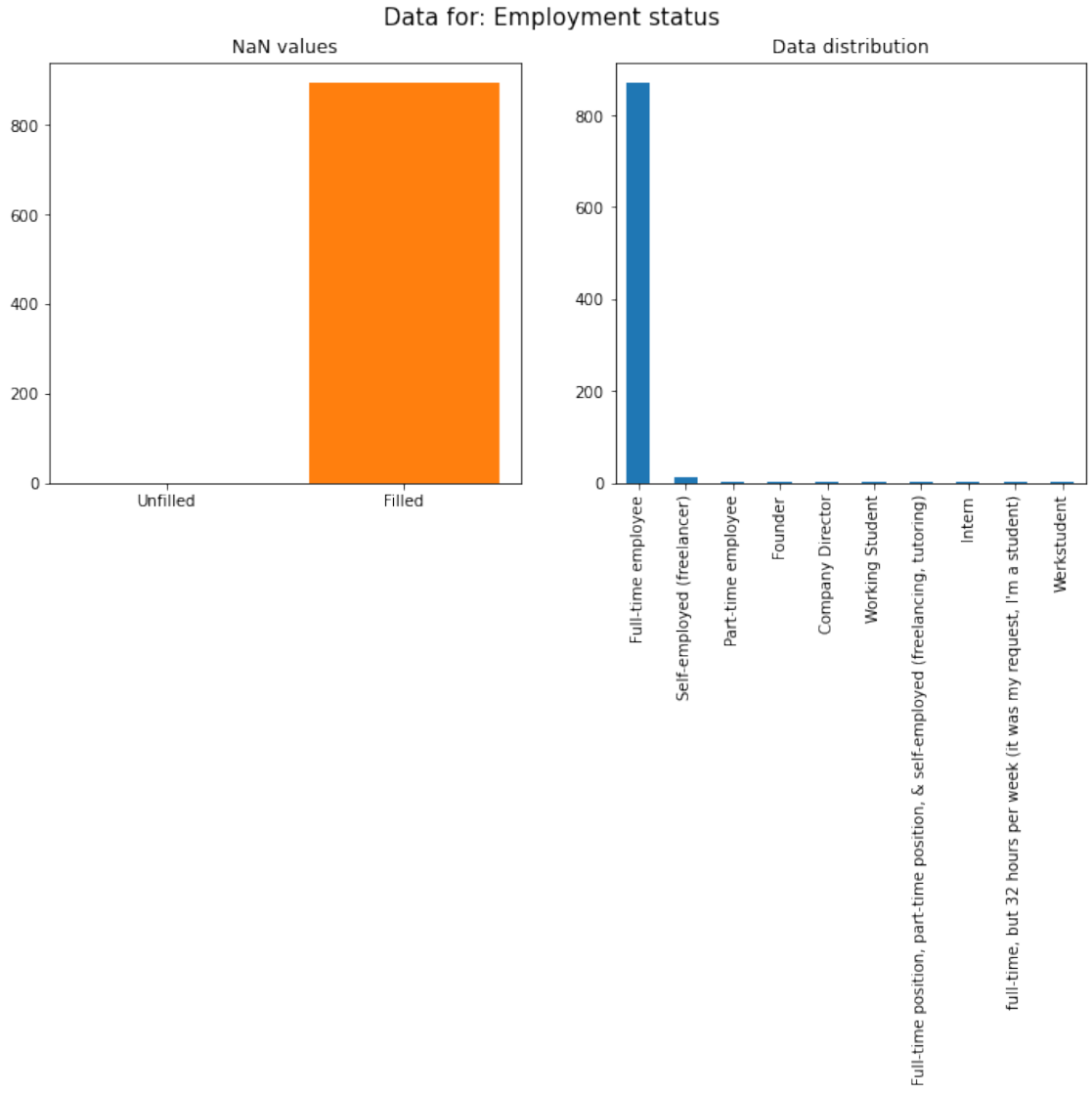

```
count       896.000000
mean       8656.660714
std       19530.577180
min           0.000000
```

```
25%          0.000000
50%          0.000000
75%       5000.000000
max      84000.000000
Name: Bonus and stocks in same country, dtype: float64
```

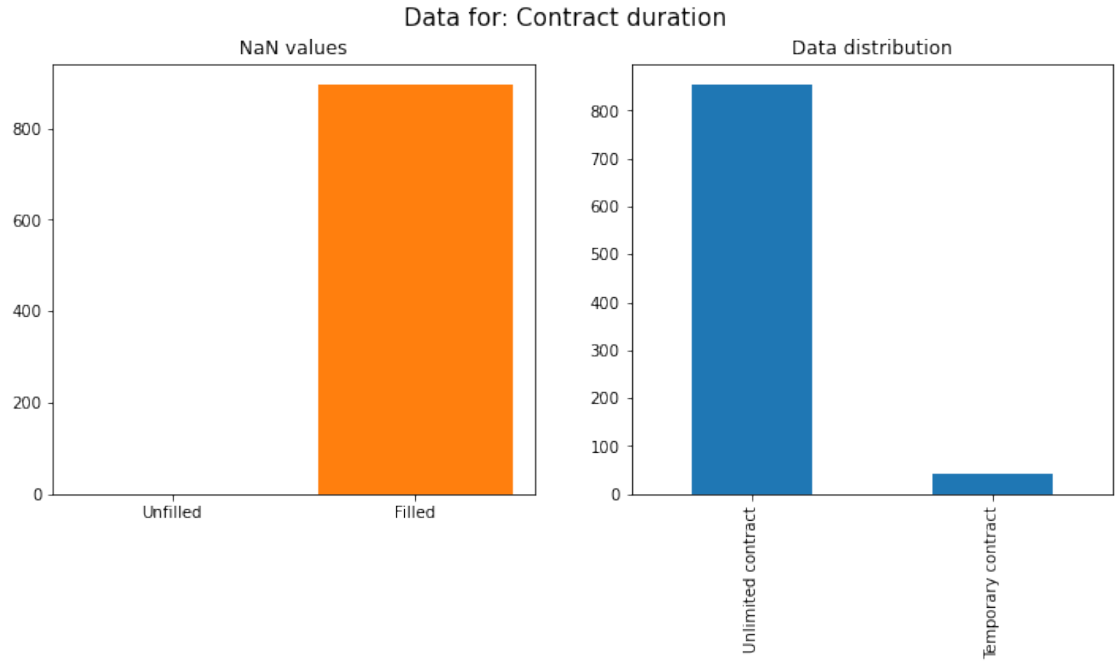

```
count    896.000000
mean      28.244420
std        2.434728
min       20.000000
25%       27.000000
50%       29.000000
75%       30.000000
max       38.000000
Name: Number of vacation days, dtype: float64
```



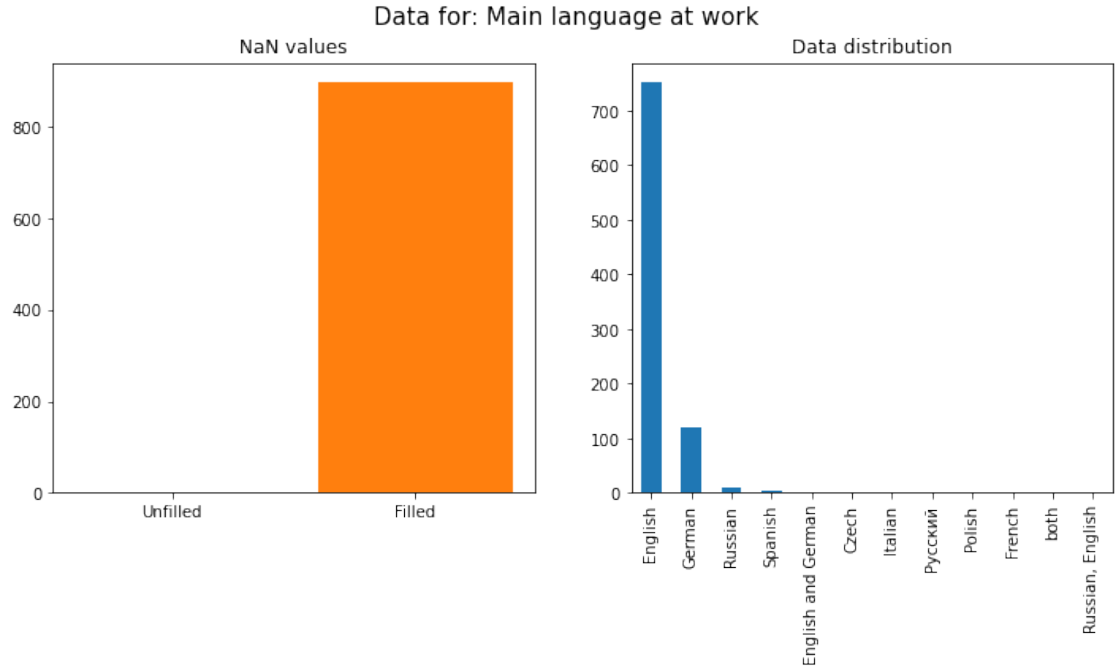

```
count                    896
unique                    10
top       Full-time employee
freq                     872
Name: Employment status, dtype: object
```




```
count                    896
unique                     2
top       Unlimited contract
freq                     854
Name:  ontract duration, dtype: object
```

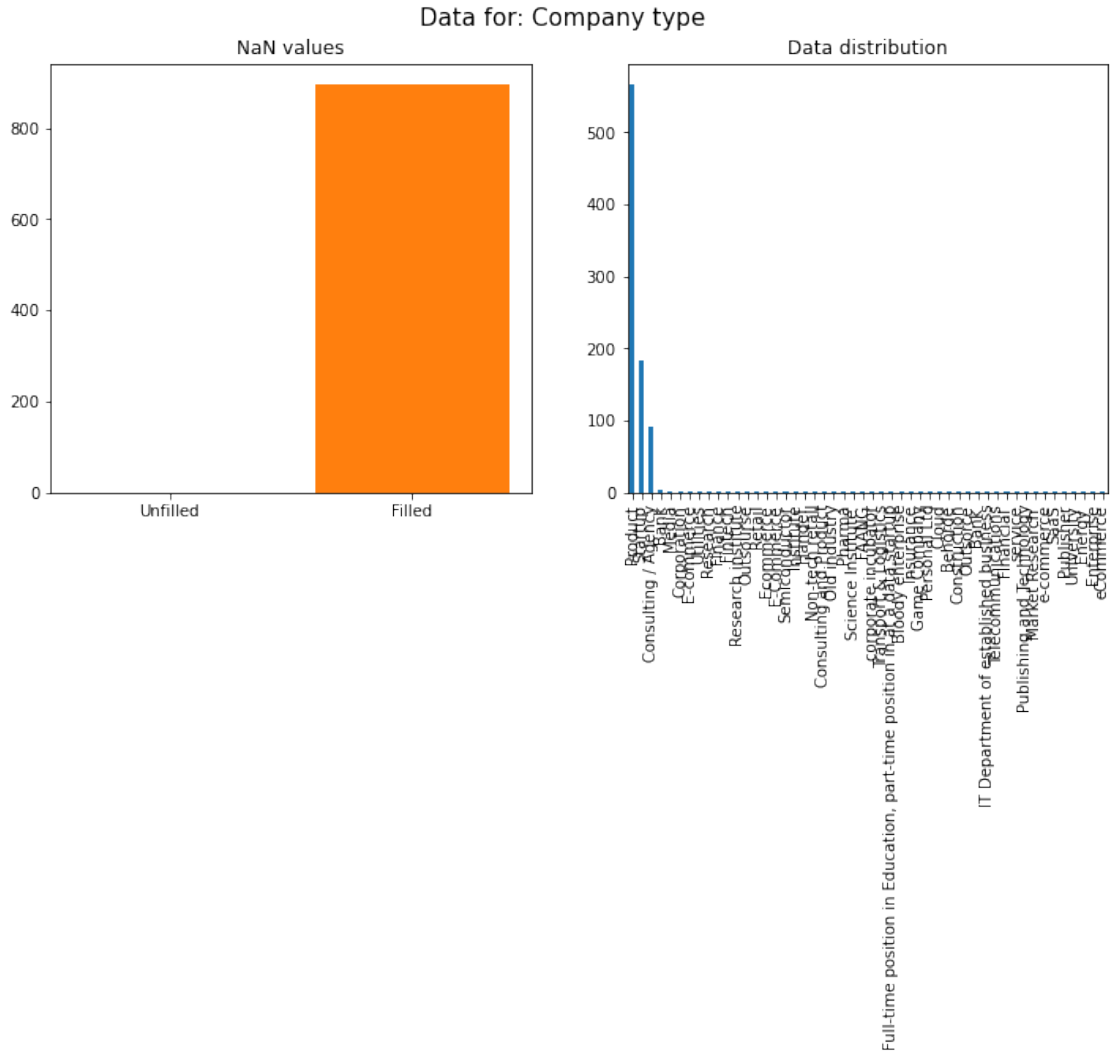



```
count          896
unique          12
top        English
freq           751
Name: Main language at work, dtype: object
```

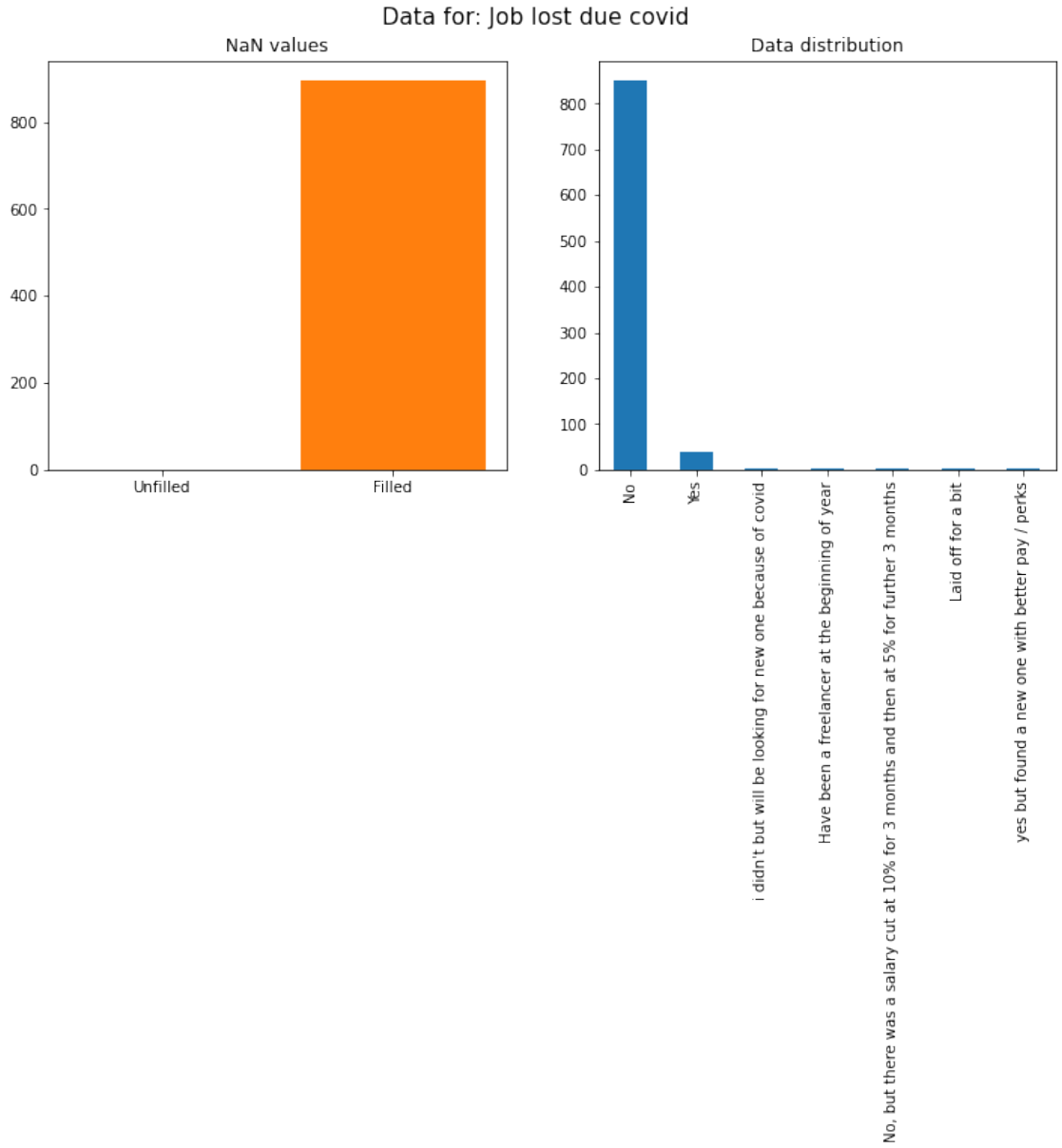


```
count       896
unique        5
top       1000+
freq        318
Name: Company size, dtype: object
```

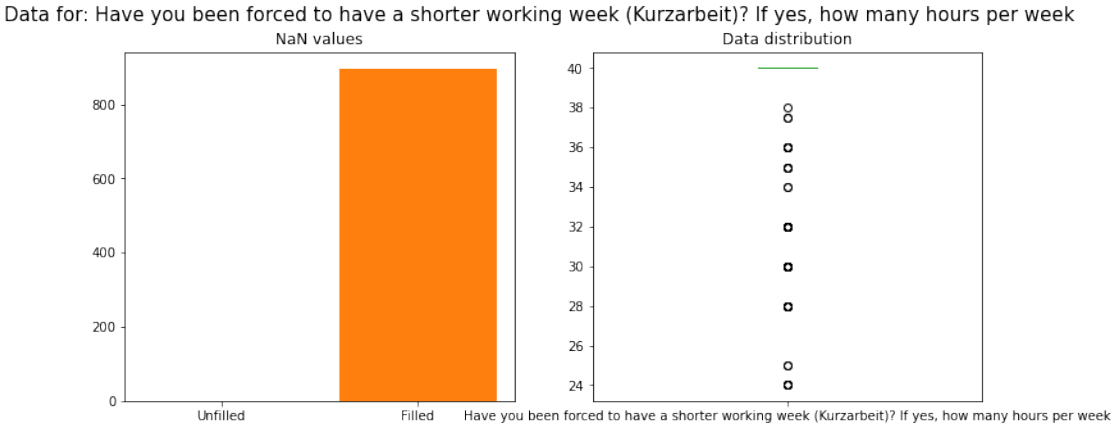

```
count         896
unique         50
top       Product
freq          567
Name: Company type, dtype: object
```



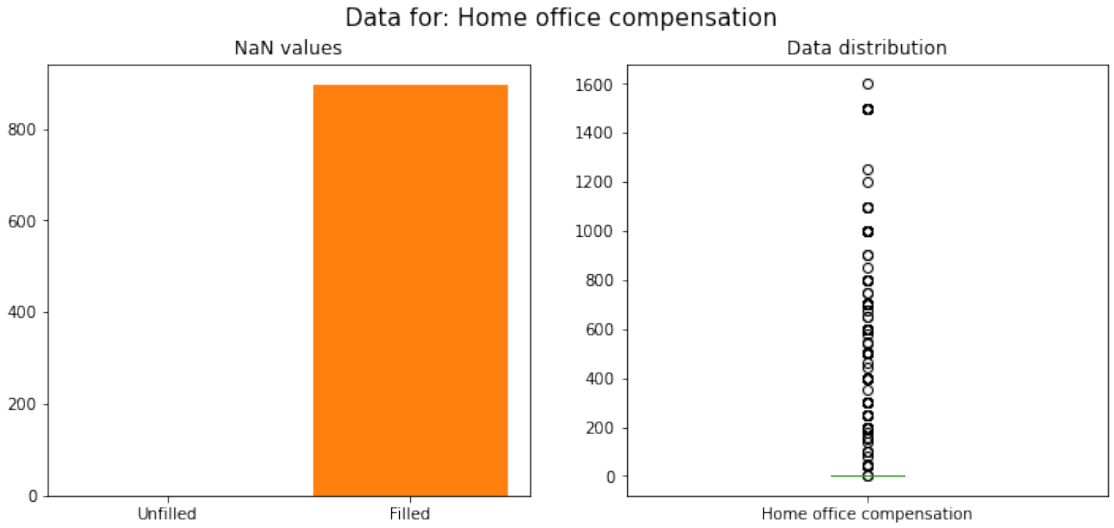

```
count     896
unique      7
top        No
freq      851
Name: Job lost due covid, dtype: object
```




```
count    896.000000
mean      39.231027
std        2.675082
min       24.000000
25%       40.000000
50%       40.000000
75%       40.000000
max       40.000000
Name: Have you been forced to have a shorter working week (Kurzarbeit)? If yes,
how many hours per week, dtype: float64
```




```
count     896.000000
mean      130.878672
std       306.620915
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max      1600.000000
Name: Home office compensation, dtype: float64
```



1.5 Predikce platu na základě ostatních parametrů

1.5.1 Převod dat pro data miningové úlohy

V této sekci si připravíme dvě datové sady, jedna bude obsahovat pouze numerické atributy a druhé pouze kategorické. Pro převod kategorického atributu na numerický použijeme kódování one hot. Tato metoda vytvoří mnoho sloupců v závislosti na počtu kategorických atributů a počtu různých hodnot. Toto kódování je vhodné pro strojové učení a například v našem případě se odhad pomocí regrese zlepšil díky tomuto kódování z 85% na 94%.

V opačném případě, když budeme připravovat kategorickou sadu, tak musíme převést numerický atribut na kateogorie. Použijeme qcut, kde pomocí kvantilů rozdělíme na určitý počet košů. Mnoho atributů jako například bonusy mají většinu hodnot rovnou nule. Zde by nešlo použit mnoho košů. Naopak jsou atributy, které klidně můžeme rozdělit do více košů jako například věk. Proto začneme u každého atributz s větším počtem košů a budeme snižovat dokud se nám nepodaří převést (Každý koš musí mít vlastní hodnotu kvantilu).

[21]:
```
nData = data2.copy(deep=True)

nData["Other technologies"] = data2["Other technologies"].str.split(pat=r"[,\/
 ↪\s]+").str.len()

nData.head()
```

[21]:

	Age	Gender	City	Position	Total years of experience	Years of experience in Germany	Seniority level	Main Technology	Other technologies	Yearly brutto	Yearly bonus + stocks in EUR	Annual brutto salary (without bonus and stocks) one year ago. Only answer if staying in the same country
0	26.0	Male	Munich	software engineer	5.0	3.0	Senior	TypeScript	3	80000.0	5000.0	75000.0
4	37.0	Male	Berlin	backend developer	17.0	6.0	Senior	C# .NET	4	62000.0	0.0	62000.0
5	32.0	Male	Berlin	devops	5.0	1.0	Senior	AWS, GCP, Python,K8s	6	76000.0	5000.0	
7	24.0	Male	Berlin	frontend developer	5.0	1.0	Senior	Typescript	2	65000.0	0.0	
8	29.0	Male	Berlin	backend developer	8.0	2.0	Senior	PHP	3	56000.0	0.0	

```
5                                          76000.0
7                                          65000.0
8                                          55000.0

   Bonus and stocks in same country  Number of vacation days  \
0                           10000.0                     30.0
4                               0.0                     29.0
5                            5000.0                     30.0
7                               0.0                     27.0
8                               0.0                     28.0

    Employment status    ontract duration Main language at work Company size  \
0  Full-time employee  Unlimited contract                English       51-100
4  Full-time employee  Unlimited contract                English     101-1000
5  Full-time employee  Unlimited contract                English        11-50
7  Full-time employee  Unlimited contract                English        1000+
8  Full-time employee  Unlimited contract                English     101-1000

  Company type Job lost due covid  \
0      Product                 No
4      Product                 No
5      Startup                 No
7      Product                 No
8      Product                 No

   Have you been forced to have a shorter working week (Kurzarbeit)? If yes, how
many hours per week  \
0                                                40.0
4                                                40.0
5                                                40.0
7                                                40.0
8                                                30.0

   Home office compensation
0                       0.0
4                       0.0
5                       0.0
7                     600.0
8                       0.0
```

[22]:
```
attributes = ["Gender", "City", "Position", "Seniority level", "Main␣
 ↪Technology", "Employment status", " ontract duration", "Main language at␣
 ↪work", "Company size", "Company type", "Job lost due covid", "Have you been␣
 ↪forced to have a shorter working week (Kurzarbeit)? If yes, how many hours␣
 ↪per week", "Home office compensation"]

one_hot = pd.get_dummies(nData)
```

```
nData = one_hot

nData=(nData-nData.min())/(nData.max()-nData.min())

#for attr in attributes:
#    pass
#    nData[attr] = pd.Categorical(nData[attr])
#    nData[attr] = nData[attr].cat.codes


#one_hot.head()

# nData.head()
```

1.5.2 Prediction

[23]:
```
def svm(dataset, target):
    brutto = dataset[target]
    testRegrData = dataset.drop([target], axis=1)


    bruttoPredict = LinearRegression()
    bruttoPredict = bruttoPredict.fit(testRegrData.values, brutto)

    #x = np.linspace(0, 175000, 175000)
    #plt.plot(dataLearn[salaryColumnName], dataLearn['Age'], 'o')

    accDif = 0.0

    for i in range(len(testRegrData)):
        p = bruttoPredict.predict([testRegrData.iloc[i]])
        br = brutto.iloc[i]
        accDif += np.abs(p - br)

    accDif /= len(testRegrData)

    print("Average delta: ", accDif)

    sc = bruttoPredict.score(testRegrData, brutto)
    print("Accuraci: ", sc)

svm(nData, "Yearly brutto")
```

```
Average delta:  [0.01986186]
Accuraci:  0.9398583867576038
```

1.5.3 Data to cat

[24]:
```
cData = data2.copy(deep=True)

attributes = ["Age", "Total years of experience", "Years of experience in␣
 ↪Germany", "Yearly brutto", "Yearly bonus + stocks in EUR", "Annual brutto␣
 ↪salary (without bonus and stocks) one year ago. Only answer if staying in␣
 ↪the same country", "Bonus and stocks in same country", "Number of vacation␣
 ↪days", "Have you been forced to have a shorter working week (Kurzarbeit)? If␣
 ↪yes, how many hours per week", "Home office compensation"]

for attr in attributes:
    #print("attr: ", attr)
    #print("attr: ", cData[attr].describe())

    #cData[attr] = pd.qcut(cData[attr], q=1) #TODO... increase q need normalize␣
 ↪

    for q in reversed(range(5)):
        try:
            cData[attr] = pd.qcut(cData[attr], q=q).astype(str)
            print("attr: ", attr, " with q: ", q)
            break
        except:
            continue


#pd.cut
#PD.CUT(column, bins=[ ],labels=[ ])
#pd.cut(df.Age,bins=[0,2,17,65,99],labels=['Toddler/
 ↪Baby','Child','Adult','Elderly'])
#print()

#cData.head()
```

```
attr:  Age  with q:  4
attr:  Total years of experience  with q:  4
attr:  Years of experience in Germany  with q:  4
attr:  Yearly brutto  with q:  4
attr:  Yearly bonus + stocks in EUR  with q:  1
attr:  Annual brutto salary (without bonus and stocks) one year ago. Only answer
if staying in the same country  with q:  4
attr:  Bonus and stocks in same country  with q:  1
attr:  Number of vacation days  with q:  4
attr:  Have you been forced to have a shorter working week (Kurzarbeit)? If yes,
how many hours per week  with q:  1
attr:  Home office compensation  with q:  1
```

[ ]: