```
pip install pandas openpyxl matplotlib seaborn numpy
```

```
Requirement already satisfied: pandas in c:\users\91952\appdata\local\
programs\python\python311\lib\site-packages (2.2.3)
Requirement already satisfied: openpyxl in c:\users\91952\appdata\
local\programs\python\python311\lib\site-packages (3.1.5)
Requirement already satisfied: matplotlib in c:\users\91952\appdata\
local\programs\python\python311\lib\site-packages (3.10.1)
Requirement already satisfied: seaborn in c:\users\91952\appdata\
local\programs\python\python311\lib\site-packages (0.13.2)
Requirement already satisfied: numpy in c:\users\91952\appdata\local\
programs\python\python311\lib\site-packages (2.2.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
91952\appdata\local\programs\python\python311\lib\site-packages (from
pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\91952\appdata\
local\programs\python\python311\lib\site-packages (from pandas)
(2025.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\91952\
appdata\local\programs\python\python311\lib\site-packages (from
pandas) (2025.1)
Requirement already satisfied: et-xmlfile in c:\users\91952\appdata\
local\programs\python\python311\lib\site-packages (from openpyxl)
(2.0.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\91952\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\91952\appdata\
local\programs\python\python311\lib\site-packages (from matplotlib)
(0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\91952\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\91952\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in c:\users\91952\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\91952\appdata\
local\programs\python\python311\lib\site-packages (from matplotlib)
(11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\91952\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib) (3.2.1)
Requirement already satisfied: six>=1.5 in c:\users\91952\appdata\
local\programs\python\python311\lib\site-packages (from python-
dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[notice] A new release of pip available: 22.3.1 -> 25.1.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

print("Libraries loaded successfully.")
```

```
Libraries loaded successfully.
```

```
df = pd.read_excel(r"C:\Users\91952\Documents\ACADEMICS\Programming
1x\Projects\Axion Assessment\Data for Task 1.xlsx")
```

Basic overview of the data

```
df.head()
```

```
                 VIN  TRANSACTION_ID  \
0  3HCFDDE89SH220903           13021   
1  1HRFFEE8XSZ230636           13028   
2  1HYKSMRK6SZ000990           13035   
3  3HCFDFEL3SH241701           13021   
4  1HRFFHEL1RZ181474           13021   

                                 CORRECTION_VERBATIM  \
0                   REPLACED STEERING WHEEL NOW OKAY   
1  CHECKED - FOUND DTC'S U0229 - U1530 SET IN BCM...   
2  APPROVED 4.9(OLH) FOR ADDED DIAGNOSTICS WITH T...   
3                         STEERING WHEEL REPLACEMENT   
4      REPLACED STEERING MESSAGE NO LONGER DISPLAYED   

                                   CUSTOMER_VERBATIM REPAIR_DATE  \
0                        STEERING WHEEL COMING APART  2024-01-02   
1         CUSTOMER STATES HEATED STEERING WHEEL INOP  2024-01-03   
2  OWNER REPORTS: THE SUPER CRUISE BAR ON THE STE...  2024-01-04   
3  CUSTOMER STATES THE LETTERING AND FINISH ON TH...  2024-01-04   
4  C/S: CUSTOMER STATES THE SERVICE DRIVER ASSIST...  2024-01-05   

                  CAUSAL_PART_NM
GLOBAL_LABOR_CODE_DESCRIPTION  \
0      WHEEL ASM-STRG *JET BLACK                Steering Wheel
Replacement   
1    MODULE ASM-STRG WHL HT CONT  Heated Steering Wheel Module
Replacement   
2  WHEEL ASM-STRG *BACKEN BLACKK                Steering Wheel
Replacement   
3      WHEEL ASM-STRG *JET BLACK                Steering Wheel
Replacement   
```

```
4      WHEEL ASM-STRG *JET BLACK                Steering Wheel
Replacement

           PLATFORM      BODY_STYLE       VPPC  ...
TRANSMISSION_TRACE_NBR  \
0  Full-Size Trucks        Crew Cab      T1CCF  ...
S2210121CNJX0941
1  Full-Size Trucks        Crew Cab      T1CGF  ...
R2210881CNJX0287
2               BEV  4 Door Utility  L233-LSOP  ...
NaN
3  Full-Size Trucks        Crew Cab      T1CCF  ...
S1210822CKJX0291
4  Full-Size Trucks        Crew Cab      T1CGF  ...
R2212982CKJX0282

   SRC_TXN_ID SRC_VER_NBR TRANSACTION_CNTR MEDIA_FLAG VIN_MODL_DESGTR
\
0  2808908219          6                1          N         CF10543

1  2808841910          6                1          Y         TF10543

2  2809979441          4                1          Y           6MB26

3  2808892288          6                1          Y         CF10543

4  2808901882          8                1          N         TF10543


  LINE_SERIES LAST_KNOWN_DELVRY_TYPE_CD  NON_CAUSAL_PART_QTY
SALES_REGION_CODE
0        1500                      21.0                    0
1
1        1500                      10.0                    0
1
2       Lux-1                      10.0                    0
1
3        1500                      10.0                    0
1
4        1500                      10.0                    0
1

[5 rows x 52 columns]
```

Shape, Column Names and Types

```
print("Shape of the dataset:", df.shape)
print("\nColumn names:")
print(df.columns.tolist())
```

```
df.info()

Shape of the dataset: (100, 52)

Column names:
['VIN', 'TRANSACTION_ID', 'CORRECTION_VERBATIM', 'CUSTOMER_VERBATIM',
'REPAIR_DATE', 'CAUSAL_PART_NM', 'GLOBAL_LABOR_CODE_DESCRIPTION',
'PLATFORM', 'BODY_STYLE', 'VPPC', 'PLANT', 'BUILD_COUNTRY',
'LAST_KNOWN_DLR_NAME', 'LAST_KNOWN_DLR_CITY', 'REPAIRING_DEALER_CODE',
'DEALER_NAME', 'REPAIR_DLR_CITY', 'STATE', 'DEALER_REGION',
'REPAIR_DLR_POSTAL_CD', 'REPAIR_AGE', 'KM', 'COMPLAINT_CD_CSI',
'COMPLAINT_CD', 'VEH_TEST_GRP', 'COUNTRY_SALE_ISO',
'ORD_SELLING_SRC_CD', 'OPTN_FAMLY_CERTIFICATION',
'OPTF_FAMLY_EMISSIOF_SYSTEM', 'GLOBAL_LABOR_CODE',
'TRANSACTION_CATEGORY', 'CAMPAIGN_NBR', 'REPORTING_COST', 'TOTALCOST',
'LBRCOST', 'ENGINE', 'ENGINE_DESC', 'TRANSMISSION',
'TRANSMISSION_DESC', 'ENGINE_SOURCE_PLANT', 'ENGINE_TRACE_NBR',
'TRANSMISSION_SOURCE_PLANT', 'TRANSMISSION_TRACE_NBR', 'SRC_TXN_ID',
'SRC_VER_NBR', 'TRANSACTION_CNTR', 'MEDIA_FLAG', 'VIN_MODL_DESGTR',
'LINE_SERIES', 'LAST_KNOWN_DELVRY_TYPE_CD', 'NON_CAUSAL_PART_QTY',
'SALES_REGION_CODE']
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 52 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   VIN                            100 non-null    object
 1   TRANSACTION_ID                 100 non-null    int64
 2   CORRECTION_VERBATIM            100 non-null    object
 3   CUSTOMER_VERBATIM              100 non-null    object
 4   REPAIR_DATE                    100 non-null    datetime64[ns]
 5   CAUSAL_PART_NM                 95 non-null     object
 6   GLOBAL_LABOR_CODE_DESCRIPTION  100 non-null    object
 7   PLATFORM                       100 non-null    object
 8   BODY_STYLE                     100 non-null    object
 9   VPPC                           100 non-null    object
 10  PLANT                          99 non-null     object
 11  BUILD_COUNTRY                  100 non-null    object
 12  LAST_KNOWN_DLR_NAME            100 non-null    object
 13  LAST_KNOWN_DLR_CITY            100 non-null    object
 14  REPAIRING_DEALER_CODE          100 non-null    object
 15  DEALER_NAME                    100 non-null    object
 16  REPAIR_DLR_CITY                100 non-null    object
 17  STATE                          98 non-null     object
 18  DEALER_REGION                  100 non-null    int64
 19  REPAIR_DLR_POSTAL_CD           98 non-null     object
 20  REPAIR_AGE                     100 non-null    int64
 21  KM                             100 non-null    int64
 22  COMPLAINT_CD_CSI               100 non-null    int64
 23  COMPLAINT_CD                   100 non-null    object
 24  VEH_TEST_GRP                   98 non-null     object
 25  COUNTRY_SALE_ISO               100 non-null    object
 26  ORD_SELLING_SRC_CD             100 non-null    int64
 27  OPTN_FAMLY_CERTIFICATION       90 non-null     object
 28  OPTF_FAMLY_EMISSIOF_SYSTEM     95 non-null     object
 29  GLOBAL_LABOR_CODE              100 non-null    int64
 30  TRANSACTION_CATEGORY           100 non-null    object
 31  CAMPAIGN_NBR                   0 non-null      float64
 32  REPORTING_COST                 100 non-null    float64
 33  TOTALCOST                      100 non-null    float64
 34  LBRCOST                        100 non-null    float64
 35  ENGINE                         100 non-null    object
 36  ENGINE_DESC                    100 non-null    object
 37  TRANSMISSION                   100 non-null    object
 38  TRANSMISSION_DESC              100 non-null    object
 39  ENGINE_SOURCE_PLANT            88 non-null     object
 40  ENGINE_TRACE_NBR               88 non-null     object
 41  TRANSMISSION_SOURCE_PLANT      88 non-null     float64
 42  TRANSMISSION_TRACE_NBR         88 non-null     object
 43  SRC_TXN_ID                     100 non-null    int64
 44  SRC_VER_NBR                    100 non-null    int64
 45  TRANSACTION_CNTR               100 non-null    int64
 46  MEDIA_FLAG                     100 non-null    object
 47  VIN_MODL_DESGTR                100 non-null    object
 48  LINE_SERIES                    99 non-null     object
 49  LAST_KNOWN_DELVRY_TYPE_CD      98 non-null     float64
 50  NON_CAUSAL_PART_QTY            100 non-null    int64
 51  SALES_REGION_CODE              100 non-null    int64
dtypes: datetime64[ns](1), float64(6), int64(12), object(33)
memory usage: 40.8+ KB
```

```
df.describe()
```

```
       TRANSACTION_ID          REPAIR_DATE  DEALER_REGION  REPAIR_AGE
\
count      100.000000                  100      100.00000  100.000000

mean     13036.900000  2024-01-22 07:26:24        1.09000   14.940000

min      13021.000000  2024-01-02 00:00:00        1.00000    0.000000

25%      13027.750000  2024-01-12 00:00:00        1.00000    5.000000

50%      13036.000000  2024-01-24 12:00:00        1.00000   12.000000

75%      13041.250000  2024-02-01 00:00:00        1.00000   21.000000

max      13081.000000  2024-02-07 00:00:00        4.00000   50.000000
```

```
std         12.028166                  NaN        0.51434   12.367945


                      KM  COMPLAINT_CD_CSI  ORD_SELLING_SRC_CD
GLOBAL_LABOR_CODE  \
count     100.000000             100.0          100.000000
100.000000
mean    24914.230000               0.0           24.590000
251.900000
min         3.000000               0.0           11.000000
20.000000
25%      8883.250000               0.0           13.000000
130.000000
50%     21962.000000               0.0           13.000000
130.000000
75%     35493.250000               0.0           48.000000
130.000000
max    107905.000000               0.0           72.000000
2400.000000
std     20747.078206               0.0           17.822976
546.451722

       CAMPAIGN_NBR  REPORTING_COST    TOTALCOST      LBRCOST  \
count           0.0      100.000000   100.000000   100.000000
mean            NaN      531.193200   563.321400   106.344900
min             NaN       27.690000    27.690000    20.000000
25%             NaN      305.432500   320.930000    61.855000
50%             NaN      433.970000   455.805000    78.560000
75%             NaN      554.062500   604.695000   108.055000
max             NaN     2457.450000  3205.450000  1012.670000
std             NaN      411.161608   456.075552   113.223074

       TRANSMISSION_SOURCE_PLANT    SRC_TXN_ID  SRC_VER_NBR
TRANSACTION_CNTR  \
count               8.800000e+01  1.000000e+02   100.000000
100.0
mean                1.676383e+08  2.815767e+09     5.720000
1.0
min                 2.878270e+05  2.808842e+09     2.000000
1.0
25%                 2.878270e+05  2.809436e+09     4.000000
1.0
50%                 8.042172e+06  2.820097e+09     4.000000
1.0
75%                 1.774929e+07  2.820880e+09     6.000000
1.0
max                 8.282984e+08  2.823000e+09    26.000000
1.0
std                 3.237538e+08  5.790727e+06     4.040402
0.0
```

```
       LAST_KNOWN_DELVRY_TYPE_CD  NON_CAUSAL_PART_QTY
SALES_REGION_CODE
count                  98.000000           100.000000
100.00000
mean                   14.132653             0.070000
1.09000
min                    10.000000             0.000000
1.00000
25%                    10.000000             0.000000
1.00000
50%                    10.000000             0.000000
1.00000
75%                    16.000000             0.000000
1.00000
max                    37.000000             1.000000
4.00000
std                     6.694570             0.256432
0.51434
```

Check Missing Values and Unique Entries

```python
for col in df.columns:
    print(f"--- {col} ---")
    print(f"Missing: {df[col].isnull().sum()}")
    print(f"Unique: {df[col].nunique()}")
    print()
```

```
--- VIN ---
Missing: 0
Unique: 98

--- TRANSACTION_ID ---
Missing: 0
Unique: 32

--- CORRECTION_VERBATIM ---
Missing: 0
Unique: 93

--- CUSTOMER_VERBATIM ---
Missing: 0
Unique: 100

--- REPAIR_DATE ---
Missing: 0
Unique: 29

--- CAUSAL_PART_NM ---
Missing: 5
```

```
Unique: 18

--- GLOBAL_LABOR_CODE_DESCRIPTION ---
Missing: 0
Unique: 4

--- PLATFORM ---
Missing: 0
Unique: 11

--- BODY_STYLE ---
Missing: 0
Unique: 6

--- VPPC ---
Missing: 0
Unique: 26

--- PLANT ---
Missing: 1
Unique: 11

--- BUILD_COUNTRY ---
Missing: 0
Unique: 3

--- LAST_KNOWN_DLR_NAME ---
Missing: 0
Unique: 100

--- LAST_KNOWN_DLR_CITY ---
Missing: 0
Unique: 94

--- REPAIRING_DEALER_CODE ---
Missing: 0
Unique: 95

--- DEALER_NAME ---
Missing: 0
Unique: 100

--- REPAIR_DLR_CITY ---
Missing: 0
Unique: 93

--- STATE ---
Missing: 2
Unique: 39
```

```
--- DEALER_REGION ---
Missing: 0
Unique: 2

--- REPAIR_DLR_POSTAL_CD ---
Missing: 2
Unique: 92

--- REPAIR_AGE ---
Missing: 0
Unique: 35

--- KM ---
Missing: 0
Unique: 100

--- COMPLAINT_CD_CSI ---
Missing: 0
Unique: 1

--- COMPLAINT_CD ---
Missing: 0
Unique: 7

--- VEH_TEST_GRP ---
Missing: 2
Unique: 23

--- COUNTRY_SALE_ISO ---
Missing: 0
Unique: 6

--- ORD_SELLING_SRC_CD ---
Missing: 0
Unique: 7

--- OPTN_FAMLY_CERTIFICATION ---
Missing: 10
Unique: 3

--- OPTF_FAMLY_EMISSIOF_SYSTEM ---
Missing: 5
Unique: 8

--- GLOBAL_LABOR_CODE ---
Missing: 0
Unique: 4

--- TRANSACTION_CATEGORY ---
Missing: 0
```

```
Unique: 2

--- CAMPAIGN_NBR ---
Missing: 100
Unique: 0

--- REPORTING_COST ---
Missing: 0
Unique: 100

--- TOTALCOST ---
Missing: 0
Unique: 100

--- LBRCOST ---
Missing: 0
Unique: 99

--- ENGINE ---
Missing: 0
Unique: 12

--- ENGINE_DESC ---
Missing: 0
Unique: 12

--- TRANSMISSION ---
Missing: 0
Unique: 19

--- TRANSMISSION_DESC ---
Missing: 0
Unique: 20

--- ENGINE_SOURCE_PLANT ---
Missing: 12
Unique: 9

--- ENGINE_TRACE_NBR ---
Missing: 12
Unique: 88

--- TRANSMISSION_SOURCE_PLANT ---
Missing: 12
Unique: 6

--- TRANSMISSION_TRACE_NBR ---
Missing: 12
Unique: 88
```

```
--- SRC_TXN_ID ---
Missing: 0
Unique: 100

--- SRC_VER_NBR ---
Missing: 0
Unique: 10

--- TRANSACTION_CNTR ---
Missing: 0
Unique: 1

--- MEDIA_FLAG ---
Missing: 0
Unique: 2

--- VIN_MODL_DESGTR ---
Missing: 0
Unique: 41

--- LINE_SERIES ---
Missing: 1
Unique: 22

--- LAST_KNOWN_DELVRY_TYPE_CD ---
Missing: 2
Unique: 11

--- NON_CAUSAL_PART_QTY ---
Missing: 0
Unique: 2

--- SALES_REGION_CODE ---
Missing: 0
Unique: 2
```

View all column names with nulls only

```
null_cols = df.columns[df.isnull().any()]
df[null_cols].isnull().sum()
```

```
CAUSAL_PART_NM                   5
PLANT                            1
STATE                            2
REPAIR_DLR_POSTAL_CD             2
VEH_TEST_GRP                     2
OPTN_FAMLY_CERTIFICATION        10
OPTF_FAMLY_EMISSIOF_SYSTEM       5
CAMPAIGN_NBR                   100
```

```
ENGINE_SOURCE_PLANT            12
ENGINE_TRACE_NBR               12
TRANSMISSION_SOURCE_PLANT      12
TRANSMISSION_TRACE_NBR         12
LINE_SERIES                     1
LAST_KNOWN_DELVRY_TYPE_CD       2
dtype: int64
```

Drop rows where Transaction ID values (key identifier) is missing

```
df = df.dropna(subset=['TRANSACTION_ID'])
```

Convert All Columns to Lowercase

```
df.columns = df.columns.str.strip().str.lower()
```

Fill Nulls Only for Columns That Exist

```
df.columns.tolist()
```

```
['vin',
 'transaction_id',
 'correction_verbatim',
 'customer_verbatim',
 'repair_date',
 'causal_part_nm',
 'global_labor_code_description',
 'platform',
 'body_style',
 'vppc',
 'plant',
 'build_country',
 'last_known_dlr_name',
 'last_known_dlr_city',
 'repairing_dealer_code',
 'dealer_name',
 'repair_dlr_city',
 'state',
 'dealer_region',
 'repair_dlr_postal_cd',
 'repair_age',
 'km',
 'complaint_cd_csi',
 'complaint_cd',
 'veh_test_grp',
 'country_sale_iso',
 'ord_selling_src_cd',
 'optn_famly_certification',
 'optf_famly_emissiof_system',
 'global_labor_code',
 'transaction_category',
 'campaign_nbr',
 'reporting_cost',
 'totalcost',
 'lbrcost',
 'engine',
 'engine_desc',
 'transmission',
 'transmission_desc',
 'engine_source_plant',
 'engine_trace_nbr',
 'transmission_source_plant',
 'transmission_trace_nbr',
 'src_txn_id',
 'src_ver_nbr',
 'transaction_cntr',
 'media_flag',
 'vin_modl_desgtr',
 'line_series',
 'last_known_delvry_type_cd',
 'non_causal_part_qty',
 'sales_region_code']

minor_null_cols = [
    'causal_part_nm', 'plant', 'state', 'repair_dlr_postal_cd',
    'veh_test_grp', 'optn_famly_certification',
    'optf_famly_emissiof_system', 'engine_source_plant',
    'engine_trace_nbr', 'transmission_source_plant',
    'transmission_trace_nbr', 'line_series',
'last_known_delvry_type_cd'
]

# Fill "unknown" only if column exists
for col in minor_null_cols:
    if col in df.columns:
        df[col] = df[col].fillna("unknown")
    else:
        print(f" Column not found: {col}")

df.rename(columns={'causal_part_nm': 'component'}, inplace=True)
```

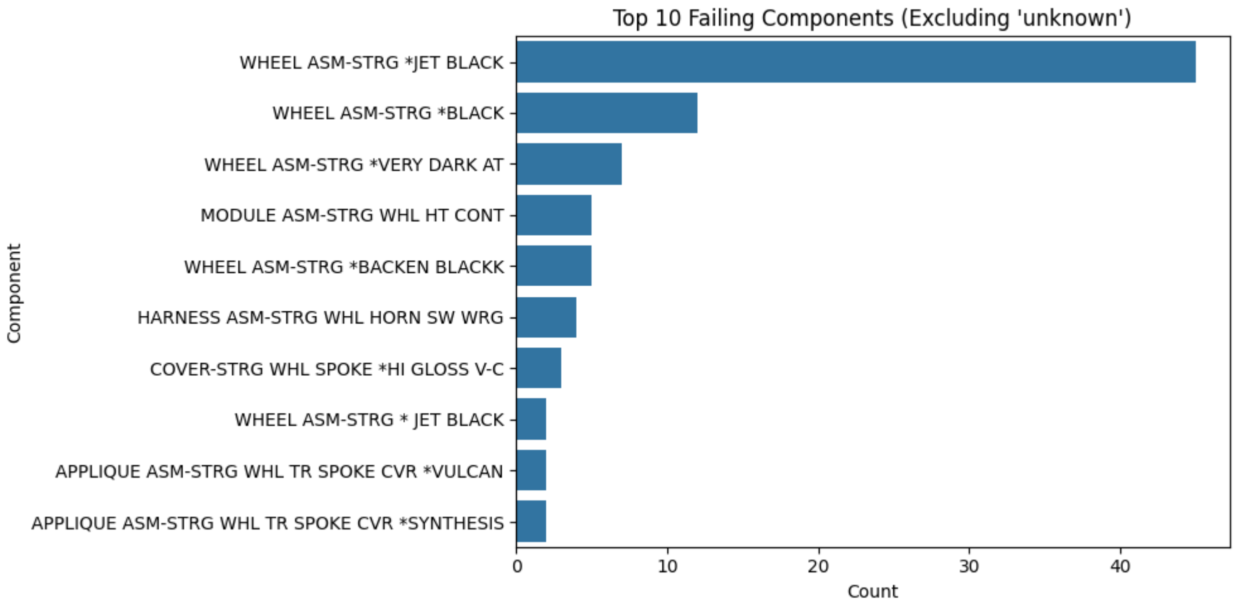
Identify Critical Columns & Create Visualizations

Column	Reason
`component`	Which part/component is failing
`repair_date`	Timeline of repairs
`totalcost` or `reporting_cost`	Cost of failure (financial impact)
`actual_hrs` or `repair_age`	Time impact of the failure
`correction_verbatim`	Root cause or fix summary (for tag extraction)

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Top 10 failing components

```
plt.figure(figsize=(10, 5))
sns.countplot(
    data=df[df['component'] != 'unknown'],
    y='component',
    order=df[df['component'] != 'unknown']
['component'].value_counts().head(10).index
)
plt.title("Top 10 Failing Components (Excluding 'unknown')")
plt.xlabel("Count")
plt.ylabel("Component")
plt.tight_layout()
plt.show()
```



Records labeled as 'unknown' represent missing or unrecorded component data and were excluded from core visual analysis to maintain the accuracy and actionability of insights.

Investigate 'unknown' Root Causes

```
df[df['component'] == 'unknown'][['vin', 'repair_date',
'correction_verbatim']].head(10)
```

```
                  vin repair_date  \
8   1HRF9CED6NZ221061  2024-01-08
19  1HR49WEY2NF322460  2024-01-11
36  3HCFDFED5SH364828  2024-01-16
47  1HCRYDED2NZ208869  2024-01-23
```

```
98  1HC4WLE78RF260518  2024-02-07

                                  correction_verbatim
8   TECHNICIAN FOUND THE STEERING COLUMN PLASTIC T...
19            Reconnected horn checked operation, ok.
36  REMOVED AND REPLACED HEATED STEERING WHEEL MOD...
47  TEST DROVE AND CONFIRMED CUSTOMER COMPLAINT BR...
98  REMOVED STEERING WHEEL AND DISASSEMBLED AND FO...
```

Group by Key Identifiers ('plant', 'dealer_name')

```
df[df['component'] == 'unknown'].groupby('plant').size().sort_values(ascending=False)

plant
FLT    2
FTW    2
SIL    1
dtype: int64

df[df['component'] == 'unknown'].groupby('dealer_name').size().sort_values(ascending=False)

dealer_name
BlueSky Auto Sales    1
IronClad Wheels       1
Legacy Car Sales      1
NovaCar Sales         1
WestPoint Motors      1
dtype: int64
```

Missing Component Analysis:

A small portion of records 6% had missing values in the `component` field, labeled as `'unknown'`. Further investigation revealed the following:

Key Findings:

- All missing entries originated from just **three plants**: `FLT`, `FTW`, and `SIL` (2 records each).
- Only **five dealers** were associated with these cases — each contributing just **1 record**.
- The free-text field `correction_verbatim` still provided clues about the actual component involved (e.g., "steering wheel", "horn").

Possible Reasons:

- System-level data mapping failure at the plant level.
- Repair submission forms allowing component fields to be skipped.
- Lack of validation in structured data capture.

Suggestions:

- Add mandatory field validation for `component` during data entry.

- Implement simple keyword-based auto-tagging from `correction_verbatim`.
- Review repair workflows or systems at plants `FLT`, `FTW`, and `SIL` to close the data gap.

These steps will help reduce data quality issues and improve insight reliability for future analyses.