Tugas Mandiri
Praktik Aggregate, Basic & Advanced Statistics

(Sesi: Advanced Statistics)

Nama : Filbert Leonardo
Kelompok : 3

1. Hitunglah semua **Measures of Central Tendency** untuk kolom *payAmount* untuk seluruh **transaksi berbayar** (tidak gratis)!

```
[45] # Filter DataFrame untuk transaksi berbayar
     paid = df[df['payAmount'] > 0]

     # Menghitung Measures of Central Tendency untuk kolom 'payAmount' pada transaksi berbayar
     mean_payAmount = paid['payAmount'].mean()
     mode_payAmount = paid['payAmount'].mode().iloc[0]
     median_payAmount = paid['payAmount'].median()

     print("Mean:", mean_payAmount)
     print("Modus:", mode_payAmount)
     print("Median:", median_payAmount)
```

```
Mean: 4919.757964929612
Modus: 3500.0
Median: 3500.0
```

2. Buatlah satu kolom baru *umur* yang berisi umur penumpang di **tahun 2024** ini. Dengan menggunakan **groupby** & **aggregate function** atau **pivot table**, hitunglah **mean** dan **median** *payAmount* dan *umur* berdasarkan kelompok *payCardBank* yang digunakan!

```
[46] # Menghitung usia dengan mengurangkan tahun kelahiran dari tahun yang diberikan
     year = 2024
     df['umur'] = year - df['payCardBirthDate']

     # Menghitung mean dan median payAmount dan umur berdasarkan kelompok payCardBank
     result = df.groupby('payCardBank').agg({'payAmount': ['median', 'mean'], 'umur': ['median', 'mean']})

     result
```

	payAmount		umur	
	median	mean	median	mean
payCardBank				
bni	3500.0	2552.278820	34.0	31.634550
brizzi	3500.0	3507.548845	34.0	34.010479
dki	3500.0	2444.656281	34.0	34.413648
emoney	3500.0	3051.308745	32.0	32.621177
flazz	3500.0	2854.880414	34.0	34.598330
online	0.0	2514.296188	35.0	34.920878

3. Buatlah beberapa kolom baru:
    - *num_stop* untuk menghitung **jumlah pemberhentian** (stopEndSeq - stopStartSeq)
    - *duration* untuk menghitung **lama perjalanan** (tapInTime-tapOutTime)

    Kemudian gunakan method **corr()** pada dataframe untuk menghitung **pearson correlation** antara *num_stop* dan *payAmount* kemudian berikan penjelasan dari hasil nilai correlation tersebut

```
[47] # Hitung jumlah berhenti untuk setiap transaksi
     df['num_stop'] = df['stopEndSeq'] - df['stopStartSeq']

     # Mengubah kolom tapInTime dan tapOutTime menjadi tipe datetime
     df['tapInTime'] = pd.to_datetime(df['tapInTime'])
     df['tapOutTime'] = pd.to_datetime(df['tapOutTime'])

     # Menghitung durasi perjalanan (menit)
     df['duration'] = (df['tapOutTime'] - df['tapInTime']).dt.total_seconds() / 60

     # Menghitung korelasi Pearson antara 'num_stop' dan 'duration_minute'
     correlation = df[['num_stop', 'duration']].corr(method='pearson')

     correlation
```

	num_stop	duration
num_stop	1.000000	0.006554
duration	0.006554	1.000000

Dalam konteks skenario hipotesis, kita dapat merumuskan sebagai berikut:

- H0 (Hipotesis Nol): Tidak ada hubungan linier yang signifikan antara jumlah berhenti dan durasi perjalanan, artinya korelasi antara 'num_stop' dan 'duration' adalah nol.
- H1 (Hipotesis Alternatif): Terdapat hubungan linier yang signifikan antara jumlah berhenti dan durasi perjalanan, artinya korelasi antara 'num_stop' dan 'duration' bukanlah nol.

Hasil analisis menunjukkan bahwa nilai korelasi sebesar 0.006554 sangat mendekati nol, yang mendukung H0. Dengan demikian, berdasarkan data yang tersedia, kita tidak memiliki bukti yang cukup untuk menolak hipotesis nol dan menyimpulkan bahwa tidak ada hubungan linier yang signifikan antara jumlah berhenti dan durasi perjalanan.

4. Buatlah 2 dataframe baru yang berisi 600 sample data dengan kriteria berikut:
    A. Sampling menggunakan **random sample method** sample()
    B. Sampling menggunakan **stratified sampling** untuk kelompok *payCardBank* , hint : dapat menggunakan kombinasi **groupby dan lambda**

   Kemudian hitung **distribusi banyaknya jumlah transaksi (dalam percentage)** di setiap kelompok *payCardBank* untuk dataframe original (sebelum di sampling), hasil random sampling, dan hasil stratified sampling. Dan tuliskan penjelasan apa yang dapat kamu simpulkan dari hasil tersebut.

```
[48] # Membuat DataFrame dengan data acak
     np.random.seed(42)  # Mengatur seed agar hasil dapat direproduksi

     # Hasil random sampling
     df_random_sample = df.sample(n=600, random_state=1)  # Menggunakan jumlah sampel yang sama dengan ukuran DataFrame asli
     # Jumlah transaksi dalam hasil random sampling untuk setiap kelompok
     jumlah_transaksi_random = df_random_sample.groupby('payCardBank').size()

     # Stratified sampling berdasarkan kelompok 'payCardBank'
     df_stratified_sample = df.groupby('payCardBank', group_keys=False).apply(lambda x: x.sample(2))
     # Jumlah transaksi dalam hasil stratified sampling untuk setiap kelompok
     jumlah_transaksi_stratified = df_stratified_sample.groupby('payCardBank').size()

     # Menghitung persentase jumlah transaksi di setiap kelompok untuk DataFrame asli
     persentase_asli = df['payCardBank'].value_counts(normalize=True)
     # Menghitung persentase jumlah transaksi di setiap kelompok untuk hasil random sampling
     persentase_random = (jumlah_transaksi_random / df_random_sample.shape[0])
     # Menghitung persentase jumlah transaksi di setiap kelompok untuk hasil stratified sampling
     persentase_stratified = (jumlah_transaksi_stratified / df_stratified_sample.shape[0])

     # Menampilkan hasil
     print("Original Distribution:")
     print(persentase_asli)
     print("\nRandom Sample Distribution:")
     print(persentase_random)
     print("\nStratified Sample Distribution:")
     print(persentase_stratified)

     Original Distribution:
```

```
[48] print("\nRandom Sample Distribution:")
     print(persentase_random)
     print("\nStratified Sample Distribution:")
     print(persentase_stratified)

     Original Distribution:
     dki       0.494538
     emoney    0.181161
     brizzi    0.093166
     flazz     0.085330
     online    0.075699
     bni       0.070106
     Name: payCardBank, dtype: float64

     Random Sample Distribution:
     payCardBank
     bni       0.071667
     brizzi    0.096667
     dki       0.540000
     emoney    0.165000
     flazz     0.061667
     online    0.065000
     dtype: float64

     Stratified Sample Distribution:
     payCardBank
     bni       0.166667
     brizzi    0.166667
     dki       0.166667
     emoney    0.166667
     flazz     0.166667
     online    0.166667
     dtype: float64
```

Dari distribusi original, hasil random sample, dan hasil stratified sample untuk kelompok *payCardBank*, kita dapat menyimpulkan bahwa:

1. Distribusi original menunjukkan proporsi penggunaan kartu **dki (49.45%)** paling tinggi, diikuti oleh **emoney (18.12%)** dan **brizzi (9.32%)**.
2. Hasil random sample dan stratified sample memiliki distribusi yang agak berbeda dengan distribusi original, tetapi stratified sample memberikan distribusi yang lebih seimbang dengan setiap kelompok *payCardBank* memiliki **proporsi yang sama (16.67%)**.

Dengan menggunakan stratified sampling, kita dapat memastikan representasi yang lebih akurat dari setiap kelompok *payCardBank* dalam sampel.