

Techinal Test_Data Scientist

Elisabeth Christina Sari

Latihan 1

Anda memiliki dataset mengenai penjualan tiket di sebuah bioskop sepanjang tahun. Dataset ini memiliki beberapa kolom berikut:

- tanggal_film (Tanggal film ditayangkan)
- judul_film (Nama film)
- durasi_film (Durasi film dalam menit)
- kapasitas_auditorium (Kapasitas tempat duduk auditorium)
- tiket_terjual (Jumlah tiket film yang terjual)
- harga_tiket (Harga tiket)

Namun, dataset ini memiliki beberapa masalah: beberapa nilai hilang atau tidak sesuai, serta adanya duplikasi. Anda harus membersihkan dataset ini agar dapat digunakan untuk analisis lebih lanjut.

```
[ ] data=read_csv("/content/drive/My Drive/dataset/penjualan_tiket.csv")  
print(data)
```

	tanggal_film	judul_film	...	tiket_terjual	harga_tiket
0	1/1/2010	aku	...	9.0	50000.0
1	8/3/2011	kamu	...	20.0	100000.0
2	4/9/2011	dia	...	15.0	20000.0
3	1/1/2010	aku	...	9.0	50000.0
4	2/13/2013	NaN	...	35.0	5000.0
5	12/10/2011	laki	...	55.0	25000.0
6	17/10/2012	cewek	...	56.0	30000.0
7	31/01/2011	jalan	...	5.0	15000.0
8	10/1/2012	kipas	...	19.0	30000.0
9	NaN	NaN	...	NaN	NaN
10	NaN	NaN	...	NaN	NaN
11	NaN	NaN	...	NaN	NaN
12	NaN	NaN	...	NaN	NaN
13	NaN	NaN	...	NaN	NaN
14	NaN	NaN	...	NaN	NaN
15	NaN	NaN	...	NaN	NaN

[16 rows x 6 columns]

```
data.dropna() #menghapus data kosong
```

	tanggal_film	judul_film	durasi_film	kapasitas_auditorium	tiket_terjual	harga_tiket
0	1/1/2010	aku	aku	10.0	9.0	50000.0
1	8/3/2011	kamu	1 jam	20.0	20.0	100000.0
2	4/9/2011	dia	30 menit	30.0	15.0	20000.0
3	1/1/2010	aku	aku	10.0	9.0	50000.0
5	12/10/2011	laki	45 menit	60.0	55.0	25000.0
6	17/10/2012	cewek	1 jam	70.0	56.0	30000.0
7	31/01/2011	jalan	35 menit	10.0	5.0	15000.0
8	10/1/2012	kipas	40 menit	30.0	19.0	30000.0

```
data.drop_duplicates(subset="tanggal_film", keep=False) #menghapus data duplikat
```

	tanggal_film	judul_film	durasi_film	kapasitas_auditorium	tiket_terjual	harga_tiket
1	8/3/2011	kamu	1 jam	20.0	20.0	100000.0
2	4/9/2011	dia	30 menit	30.0	15.0	20000.0
4	2/13/2013	NaN	15 menit	50.0	35.0	5000.0
5	12/10/2011	laki	45 menit	60.0	55.0	25000.0
6	17/10/2012	cewek	1 jam	70.0	56.0	30000.0
7	31/01/2011	jalan	35 menit	10.0	5.0	15000.0
8	10/1/2012	kipas	40 menit	30.0	19.0	30000.0

Latihan 2

Anda diberi dataset yang mencakup informasi seputar perjalanan pelanggan selama satu bulan, seperti:

- tanggal_waktu (Tanggal dan waktu mulai perjalanan)
- jarak (Jarak perjalanan dalam kilometer)
- durasi (Durasi perjalanan dalam menit)
- harga (Harga perjalanan)
- driver_rating (Rating yang diberikan oleh pelanggan kepada pengemudi)
- customer_rating (Rating yang diberikan oleh pengemudi kepada pelanggan)

Anda diminta untuk membuat visualisasi yang efektif dari data tersebut untuk mengeksplorasi pola dan hubungan antara fitur-fiturnya.

```
[ ] df = pd.read_csv("/content/drive/My Drive/dataset/info_perjalanan.csv")  
df.head(10)
```



	tgl_waktu	jarak (km)	durasi (menit)	harga	driverrating	cust_rating
0	1/1/2016 21:17	5.1	18	22000	4.8	4.0
1	1/2/2016 1:37	5.0	15	20000	4.7	5.0
2	1/2/2016 20:38	4.8	17	19200	4.7	4.8
3	1/6/2016 17:19	4.7	16	18800	5.0	4.6
4	1/7/2016 13:33	7.0	21	28000	4.6	5.0
5	1/10/2016 8:25	0.8	3	3200	5.0	4.8
6	1/10/2016 12:44	16.5	48	66000	4.3	4.6
7	1/10/2016 19:32	7.5	23	30000	4.9	4.0
8	1/11/2016 9:21	6.2	20	24800	4.8	4.8
9	1/11/2016 12:03	6.4	24	25600	4.7	5.0

```
[ ] df.shape
```

```
(24, 6)
```

Banyak kolom dan baris pada dataset

```
[ ] df.size
```

```
144
```

```
[ ] df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 24 entries, 0 to 23  
Data columns (total 6 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   tgl_waktu        24 non-null     object  
1   jarak (km)       24 non-null     float64  
2   durasi (menit)   24 non-null     int64  
3   harga            24 non-null     int64  
4   driverrating     24 non-null     float64  
5   cust_rating      24 non-null     float64  
dtypes: float64(3), int64(2), object(1)  
memory usage: 1.2+ KB
```

Informasi type dataset

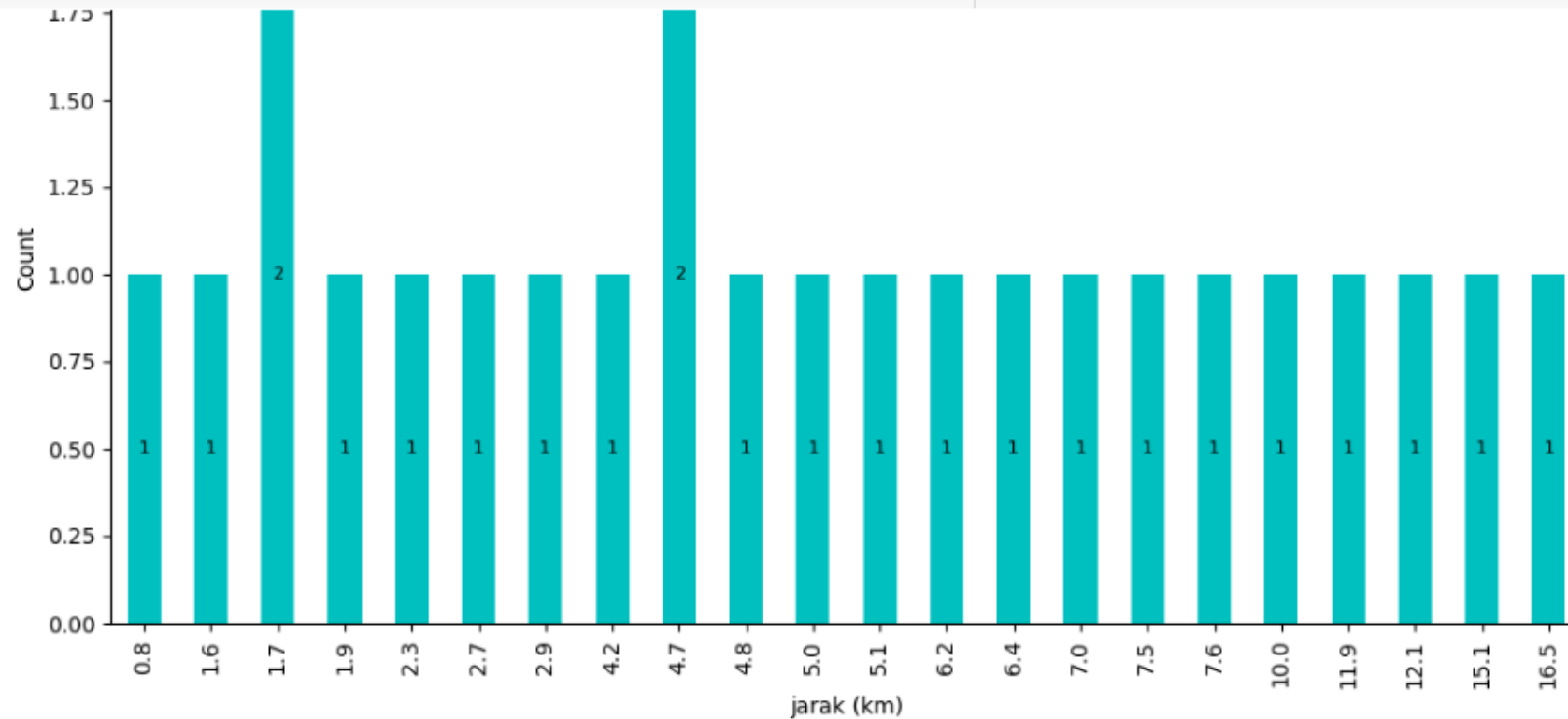
Nilai min, max, rata-rata, q1, q2, dll pada setiap kolom

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

	jarak (km)	durasi (menit)	harga	driverrating	cust_rating
count	24.000000	24.000000	24.000000	24.000000	24.000000
mean	6.016667	19.083333	24133.333333	4.737500	4.625000
std	4.330545	12.430386	17310.531267	0.255058	0.359045
min	0.800000	3.000000	3200.000000	4.000000	4.000000
25%	2.600000	9.500000	10400.000000	4.675000	4.300000
50%	4.900000	17.000000	19600.000000	4.750000	4.800000
75%	7.525000	25.000000	30100.000000	4.925000	4.900000
max	16.500000	48.000000	66000.000000	5.000000	5.000000

Jarak rata-rata yang sering muncul dalam perjalanan adalah 1,7 dan 4,7 km

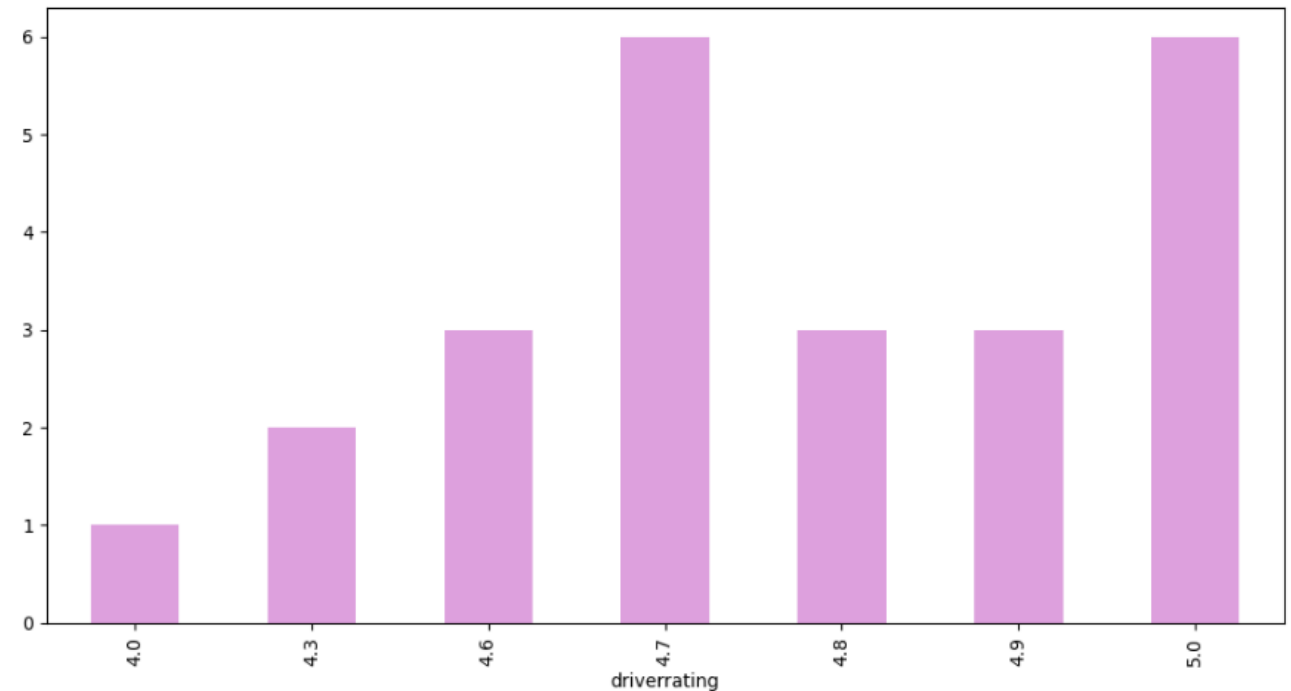
```
[ ] ax = data.groupby('jarak (km)')['durasi (menit)'].count().reset_index().plot(
    kind='bar', stacked = False, x='jarak (km)', color='c', legend = False)
plot_counts(ax)
```



```
[ ] first_option_data = data [(data['jarak (km)'] >= 5) & (data['durasi (menit)'] >= 10) &
                             (data['harga'] >= 10000) & (data['driverrating'] >= 4.7)]
first_option_data.head()
```

	tgl_waktu	jarak (km)	durasi (menit)	harga	driverrating	cust_rating
0	1/1/2016 21:17	5.1	18	22000	4.8	4.0
1	1/2/2016 1:37	5.0	15	20000	4.7	5.0
7	1/10/2016 19:32	7.5	23	30000	4.9	4.0
8	1/11/2016 9:21	6.2	20	24800	4.8	4.8
9	1/11/2016 12:03	6.4	24	25600	4.7	5.0

Rating driver yg diberikan oleh customer rata-rata masih belum memuaskan. Dari data tersebut bisa digunakan untuk memperbaiki sikap atau cara menyetir driver agar lebih memuaskan customer dalam perjalanan



Latihan 3

Anda bekerja di divisi Human Resources sebuah perusahaan dan diberi dataset yang mencakup informasi tentang karyawan, termasuk:

- umur (Umur karyawan)
- jenis_kelamin (Jenis kelamin karyawan)
- pendidikan (Tingkat pendidikan karyawan)
- lama_bekerja (Lama bekerja di perusahaan dalam tahun)
- gaji (Gaji karyawan)

Tugas Anda adalah menggunakan statistik deskriptif dan inferensial untuk menganalisis dataset ini dan memahami pola dan hubungan antara fitur-fiturnya.



```
[ ] data=read_csv("/content/drive/My Drive/dataset/HR.csv")  
print(data)
```

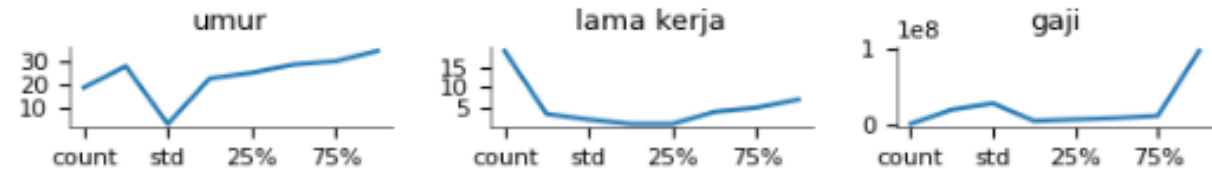
	umur	jenis_kelamin	pendidikan	lama_kerja	gaji
0	25	p	S1	1	5000000
1	24	p	D3	1	4800000
2	26	p	S1	2	6000000
3	27	l	S1	1	5500000
4	30	p	S1	6	7000000
5	25	l	D3	1	6000000
6	32	l	S2	3	18000000
7	23	l	D3	1	4000000
8	27	l	S1	4	65000000
9	28	p	S1	5	7000000
10	31	p	S1	6	9000000
11	32	p	D3	4	9200000
12	35	l	S1	7	10000000
13	29	p	S1	4	8000000
14	29	l	D3	5	8200000
15	30	l	S2	3	12000000
16	25	p	S1	1	7000000
17	29	l	S1	5	75000000
18	31	l	D3	6	98000000

Statistik deskriptif

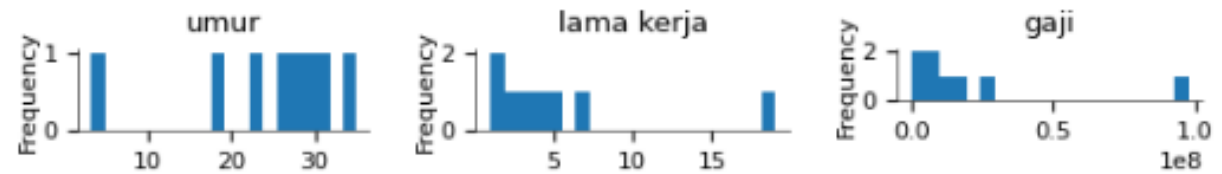
```
[ ] data.describe()
```

	umur	lama kerja	gaji
count	19.000000	19.000000	1.900000e+01
mean	28.315789	3.473684	1.919474e+07
std	3.180717	2.091475	2.751954e+07
min	23.000000	1.000000	4.000000e+06
25%	25.500000	1.000000	6.000000e+06
50%	29.000000	4.000000	8.000000e+06
75%	30.500000	5.000000	1.100000e+07
max	35.000000	7.000000	9.800000e+07

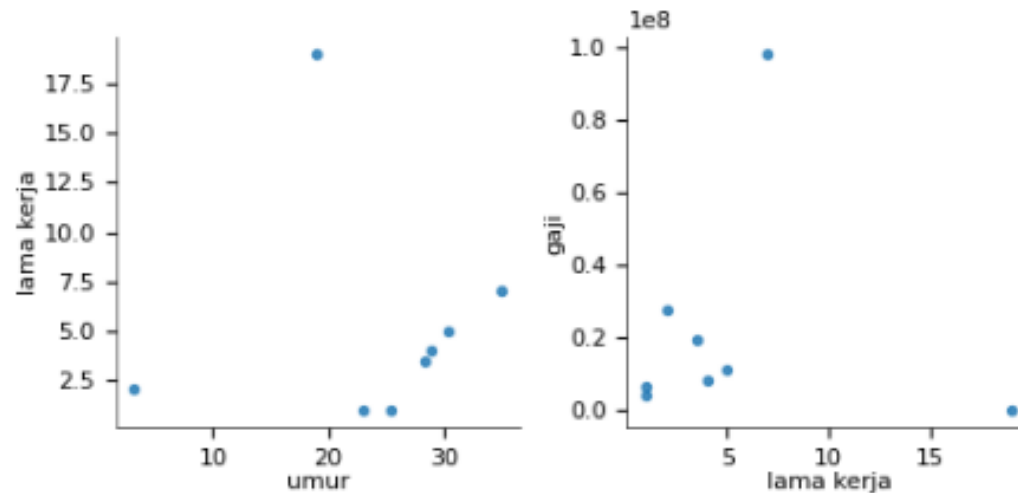
```
[ ] Values
```



Distributions



2-d distributions



```
[ ] data.describe(include=['object'])
```

	jenis kelamin	pendidikan
count	19	19
unique	2	3
top	1	S1
freq	10	11

```
[ ] data['jenis kelamin'].value_counts()
```

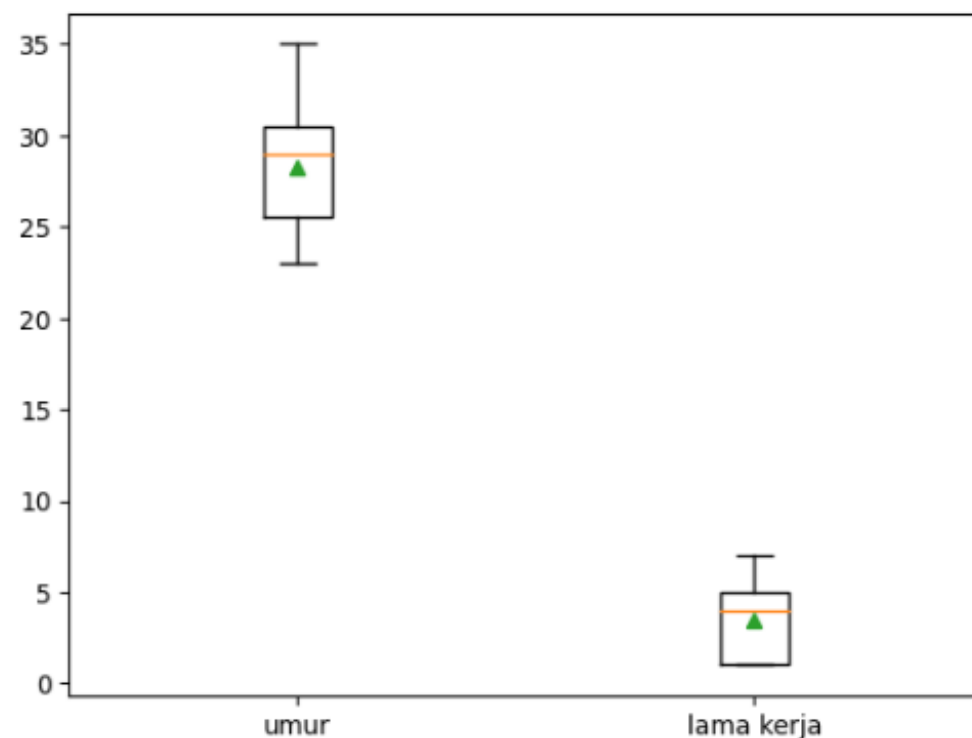
```
1    10
p     9
Name: jenis kelamin, dtype: int64
```

```
[ ] data['pendidikan'].value_counts()
```

```
S1    11
D3     6
S2     2
Name: pendidikan, dtype: int64
```

```
[ ] umur, lamakerja = data['umur'],data['lama kerja']
```

```
[ ] plt.boxplot([umur, lamakerja], labels=['umur','lama kerja'], showmeans=True);
```



Inferensial

```
[ ] population = np.random.randint(20,30,1000)
    np.random.seed(10)

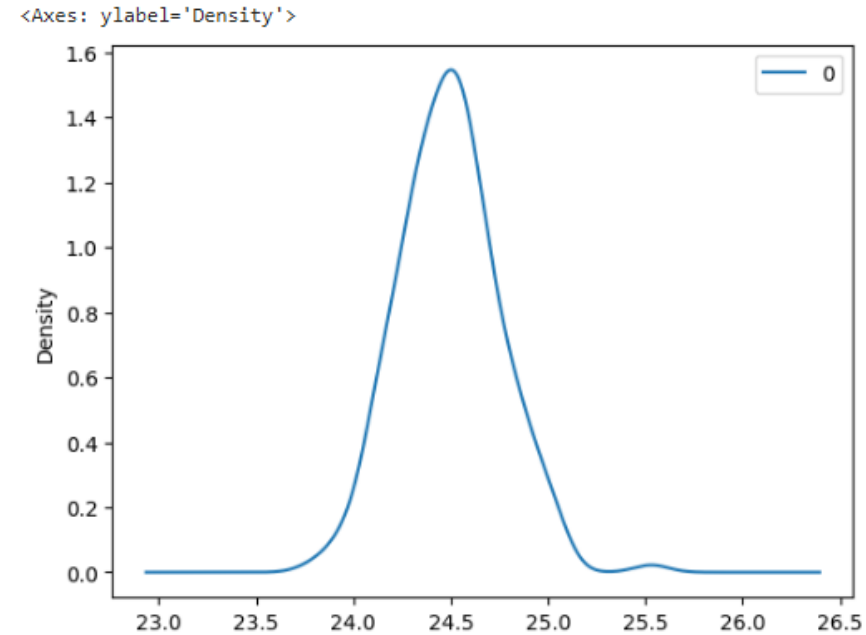
    estimates=[]

    for x in range(200):
        sample = np.random.choice(a=population, size=100)
        estimates.append (sample.mean())
```

```
[ ] np.mean(population)
```

24.458

```
[ ] pd.DataFrame(estimates).plot(kind="density")
```



```
[ ] import scipy.stats as stats
```

```
    z_critical =stats.norm.ppf(q=0.9)
```

```
[ ] t_critical =stats.t.ppf(q=0.9, df=24)
```

```
[ ] margin_of_error = z_critical * (np.std(estimates)/np.sqrt(200))
```

```
[ ] np.mean(estimates) - margin_of_error
```

24.470233463780538

Latihan 4

Anda diberi dataset mengenai pelanggan sebuah perusahaan kartu kredit. Dataset ini mencakup informasi seperti:

- pelanggan_id (ID unik untuk setiap pelanggan)
- usia (Usia pelanggan)
- jenis_kelamin (Jenis kelamin pelanggan)
- pendapatan (Pendapatan tahunan pelanggan)
- jml_kartu_kredit (Jumlah kartu kredit yang dimiliki pelanggan)
- pengeluaran_bulanan (Pengeluaran bulanan rata-rata pelanggan)

Anda diminta untuk membuat model machine learning yang akan memprediksi apakah seorang pelanggan akan tertarik pada penawaran kartu kredit baru. Anda harus menyiapkan lembar kerja yang meliputi proses pengembangan model, termasuk pemilihan fitur, pelatihan, validasi, dan evaluasi.



data



	id_pelanggan	usia	jk	pendapatan	jmlh_kartu_kredit	pengeluaran	respon
0	1	40	P	72000000	3	6000000	1
1	2	28	P	36000000	1	4500000	0
2	3	31	L	60000000	2	5000000	0
3	4	28	P	36000000	1	3000000	1
4	5	38	L	72000000	2	5000000	1
5	6	28	P	58000000	2	6200000	1
6	7	40	L	36000000	1	4000000	0
7	8	38	P	72000000	4	10000000	1
8	9	31	L	70000000	3	8000000	0
9	10	41	P	60000000	1	5300000	0
10	11	44	L	36000000	1	4500000	1
11	12	28	P	58000000	2	5000000	0
12	13	41	L	72000000	4	12000000	1

```
[ ] data.dtypes
```

```
id_pelanggan    int64
usia             int64
jk              object
pendapatan       int64
jmlh_kartu_kredit int64
pengeluaran      int64
respon           int64
dtype: object
```

```
[ ] for col in data.columns:
    if data[col].dtypes == 'object':
        print(col, data[col].unique())
```

```
jk ['P' 'L']
```

```
[ ] cols_to_label=[]
for i in data.columns:
    if data[i].dtypes == 'O':
        cols_to_label.append(i)
```

```
cols_to_label
```

```
['jk']
```

```
[ ] data[cols_to_label] = data[cols_to_label].apply(LabelEncoder().fit_transform)
```

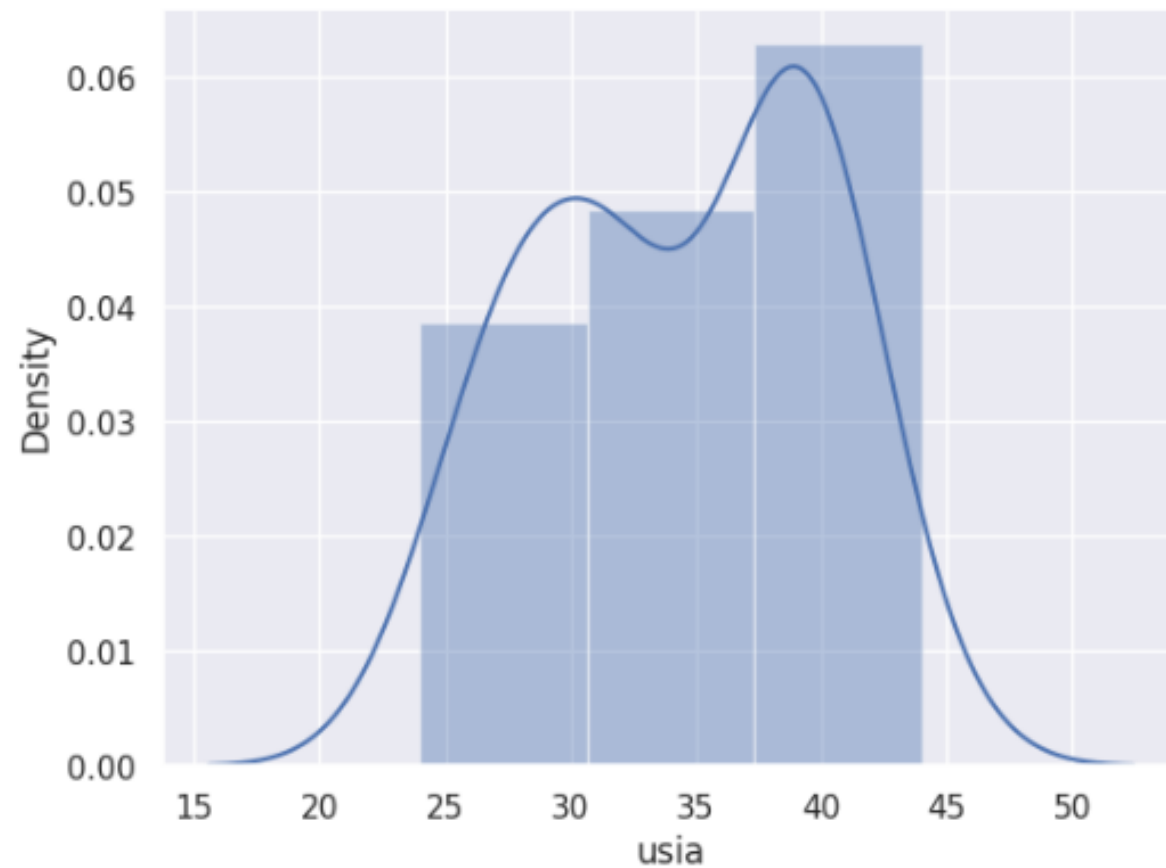
```
[ ] data.head()
```

	id_pelanggan	usia	jk	pendapatan	jmlh_kartu_kredit	pengeluaran	respon
0	1	40	1	72000000	3	6000000	1
1	2	28	1	36000000	1	4500000	0
2	3	31	0	60000000	2	5000000	0
3	4	28	1	36000000	1	3000000	1
4	5	38	0	72000000	2	5000000	1

```
[ ] data.isnull().sum()
```

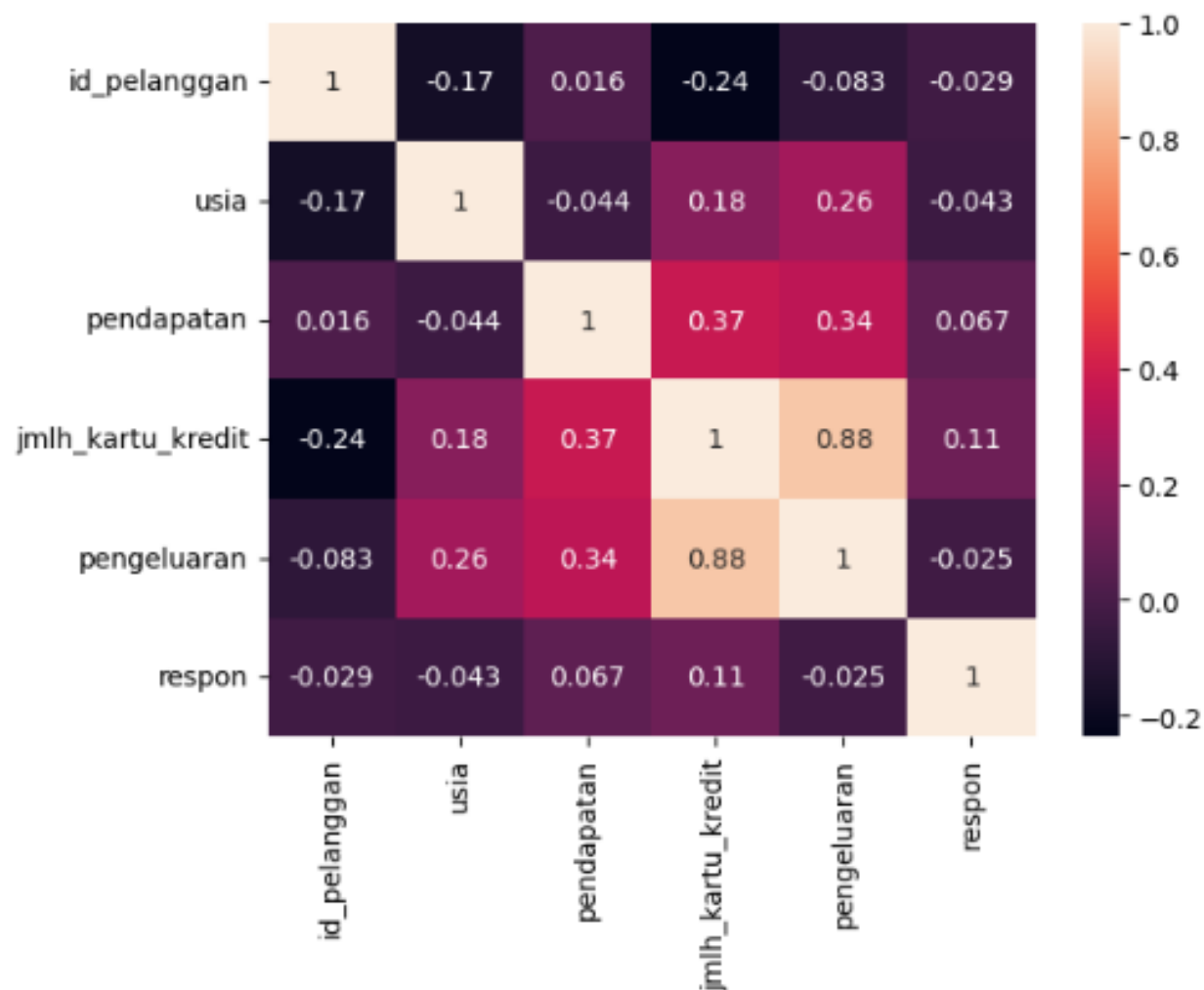
```
id_pelanggan    0  
usia            0  
jk              0  
pendapatan      0  
jmlh_kartu_kredit 0  
pengeluaran     0  
respon          0  
dtype: int64
```

```
↳ sns.distplot(data['usia'])  
<Axes: xlabel='usia', ylabel='Density'>
```



```
import seaborn as sns
sns.heatmap(data.corr(),annot=True)
```

```
<ipython-input-33-53fe516a88e7>:2: FutureWarning: The default value of numeric_only in DataFrame.
sns.heatmap(data.corr(),annot=True)
<Axes: >
```



◀

```
[ ] X_train, X_test, y_train, y_test = train_test_split(data.drop('respon', axis=1), data['respon'], test_size=0.2)
```

```
[ ] model1 = LogisticRegression(random_state = 20).fit(X_train, y_train)
    preds = model1.predict(X_test)
    print(f'The accuracy score of Logistic Regression model is: {accuracy_score(preds, y_test)}')
```

The accuracy score of Logistic Regression model is: 0.42857142857142855

```
[ ] model2 = XGBClassifier().fit(X_train, y_train)
    preds = model2.predict(X_test)
    print(f'The accuracy score of XGBClassifier model is: {accuracy_score(preds, y_test)}')
```

The accuracy score of XGBClassifier model is: 0.42857142857142855

Latihan 5

Kami telah menyediakan contoh data produk di Sephora. Dari data ini, kami ingin tahu wawasan apa yang bisa Anda dapatkan. Beberapa bagian dari wawasan harus terkait dengan model pembelajaran mesin dan visualisasi data.

Data: <https://binapertiwi.link/filesdatascientisttest>



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8000 entries, 0 to 7999
Data columns (total 17 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                     8000 non-null   int64
1   brand                  8000 non-null   object
2   category               8000 non-null   object
3   name                   8000 non-null   object
4   size                   8000 non-null   object
5   rating                 8000 non-null   float64
6   number_of_reviews      8000 non-null   int64
7   love                   8000 non-null   int64
8   price                  8000 non-null   float64
9   value_price            8000 non-null   float64
10  URL                    8000 non-null   object
11  MarketingFlags         8000 non-null   bool
12  options                 8000 non-null   object
13  details                 8000 non-null   object
14  how_to_use             8000 non-null   object
15  ingredients             8000 non-null   object
16  exclusive              8000 non-null   int64
dtypes: bool(1), float64(3), int64(4), object(9)
memory usage: 1007.9+ KB
```

```
[ ] df_sephora = df_sephora.drop(['id','size','size','URL','how_to_use','ingredients','details'],axis=1)
df_sephora.head()
```

	brand	category	name	rating	number_of_reviews	love	price	value_price	MarketingFlags	options	exclusive
0	SEPHORA COLLECTION	no category	Gift Card	5.0	46	0	50.0	50.0	False	no options	0
1	SEPHORA COLLECTION	no category	Happy Birthday Gift Card	0.0	0	0	50.0	50.0	False	no options	0
2	SEPHORA COLLECTION	no category	Lips Gift Card	0.0	0	0	50.0	50.0	False	no options	0
3	SEPHORA COLLECTION	no category	Thank You Gift Card	0.0	0	0	50.0	50.0	False	no options	0
4	SEPHORA COLLECTION	no category	Heart Gift Card	0.0	0	0	50.0	50.0	False	no options	0

```
[ ] pd.set_option('display.max_rows', None)
df_sephora.category.value_counts() #jumlah kategori produk
```

```

Perfume          620
Moisturizers     398
Face Serums      334
Value & Gift Sets 241
Face Wash & Cleansers 225
Face Masks       218
Hair Styling Products 213
Rollerballs & Travel Size 212
Face Brushes     176
Eye Creams & Treatments 171
Eye Palettes     170
Shampoo          162
Lipstick         162
Foundation       153
Highlighter      152
Cologne          148
Mascara          142
Conditioner      130
Candles & Home Scents 123
Face Primer      121
Eyeliner         111
Setting Spray & Powder 104
Makeup           104
Mini Size        95
Eye Brushes      94
Body Lotions & Body Oils 93
Lotions & Oils   92
Eyebrow          92
Beauty Supplements 92

```

```
▶ sephora_brands = df_sephora.brand.value_counts()  
sephora_brands.head(10) #brand produk terbanyak
```

```
⦿ SEPHORA COLLECTION      492  
  CLINIQUE                 211  
  TOM FORD                 150  
  tarte                    143  
  Kiehl's Since 1851      122  
  Dior                     118  
  Fresh                    108  
  Lancôme                  104  
  Bumble and bumble        99  
  MAKE UP FOR EVER         95  
  Name: brand, dtype: int64
```

```
[ ] sephora_brands.tail(10) #brand paling sedikit
```

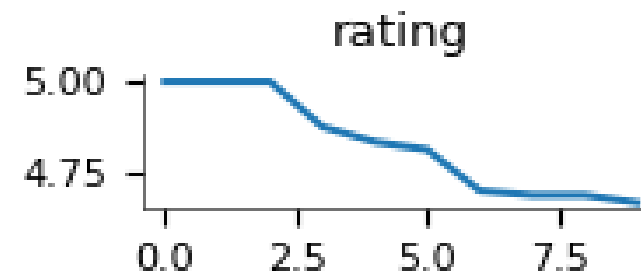
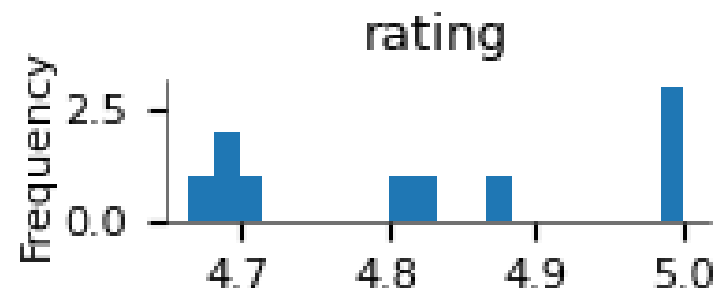
```
  8Greens                  1  
  Saint Jane Beauty        1  
  Cellu-cup                1  
  Too Cool For School      1  
  Google                   1  
  Conture                   1  
  Cocofloss                 1  
  Makeup Eraser            1  
  Flora + Bast              1  
  SHHHOWERCAP              1
```

```
[ ] rand_love=pd.DataFrame(df_sephora.groupby(['brand'])['love'].mean()).reset_index().sort_values(by='love', ascending=False).reset_index().drop('index',axis=1)
brand_love.head(10) #brand paling banyak di love
```

	brand	love
0	stila	140942.857143
1	Buxom	138700.000000
2	NARS	102575.000000
3	Anastasia Beverly Hills	92904.687500
4	Drunk Elephant	84009.090909
5	Makeup Eraser	79500.000000
6	Urban Decay	76271.428571
7	FENTY BEAUTY by Rihanna	71598.412698
8	Tatcha	66725.925926
9	The Ordinary	66062.857143

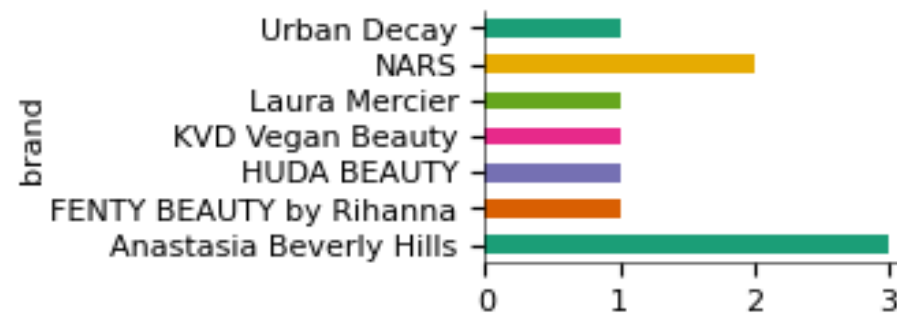
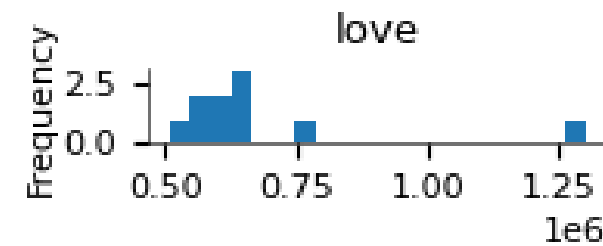
```
[ ] brand_rating=pd.DataFrame(df_sephora.groupby('brand')['rating'].mean()).reset_index().sort_values(by='rating', ascending=False).reset_index().drop('index',axis=1)
brand_rating.head(10) #rating tertinggi
```

	brand	rating
0	Montblanc	5.000000
1	Four Sigmatic	5.000000
2	Aether Beauty	5.000000
3	Golde	4.875000
4	ReFa	4.833333
5	RODIN olio lusso	4.812500
6	The Art of Shaving	4.700000
7	Paco Rabanne	4.687500
8	SOBEL SKIN Rx	4.687500
9	StackedSkincare	4.666667



```
[ ] Product_love = pd.DataFrame(df_sephora.groupby(['brand','name'])['love'].mean()).reset_index().sort_values(by='love', ascending=False).reset_index().drop('index',axis=1)
Product_love.head(10) #produk yang paling banyak disukai
```

	brand	name	love
0	KVD Vegan Beauty	Everlasting Liquid Lipstick	1300000.0
1	NARS	Radiant Creamy Concealer	770700.0
2	Anastasia Beverly Hills	Brow Wiz	660000.0
3	Laura Mercier	Translucent Loose Setting Powder	657100.0
4	NARS	Blush	646600.0
5	FENTY BEAUTY by Rihanna	Pro Filt'r Soft Matte Longwear Foundation	625500.0
6	HUDA BEAUTY	Obsessions Eyeshadow Palette	624600.0
7	Anastasia Beverly Hills	Eye Shadow Singles	565200.0
8	Anastasia Beverly Hills	Liquid Lipstick	549000.0
9	Urban Decay	All Nighter Long-Lasting Makeup Setting Spray	506800.0



Rata-rata harga skincare

```
[ ] skincare = pd.DataFrame(df_sephora.loc[df_sephora.category == 'Skincare', :].sort_values(by='rating', ascending=False))
skincare.head()
```

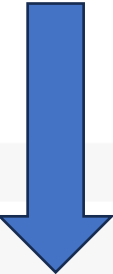
	brand	category	name	rating	number_of_reviews	love	price	value_price	MarketingFlags	options	exclusive
646	Jack Black	Skincare	Beard Lube® Conditioning Shave Mini	5.0	8	1000	11.5	11.5	True	no options	0
1799	Marvis	Skincare	Jasmin Mint Toothpaste Mini	5.0	19	3900	6.0	6.0	True	no options	0
202	CLINIQUE	Skincare	Mini Dramatically Different Moisturizing Gel	4.5	3000	14700	15.0	15.0	False	no options	0
1661	St. Tropez	Skincare	Self Tan Express Bronzing Mousse Mini	4.5	10	3600	19.0	19.0	False	no options	0
5027	Farmacy	Skincare	Mini Green Clean Makeup Meltaway Cleansing Balm	4.5	78	0	22.0	22.0	True	no options	1

```
[ ] ratahargaskincare = skincare["price"].mean()
ratahargaskincare
```

16.91304347826087

```
[ ] skincare['rata - rata harga'] = ratahargaskincare
```

```
[ ] skincare.head(5)
```



	brand	category	name	rating	number_of_reviews	love	price	value_price	MarketingFlags	options	exclusive	rata - rata harga
646	Jack Black	Skincare	Beard Lube® Conditioning Shave Mini	5.0	8	1000	11.5	11.5	True	no options	0	16.913043
1799	Marvis	Skincare	Jasmin Mint Toothpaste Mini	5.0	19	3900	6.0	6.0	True	no options	0	16.913043
202	CLINIQUE	Skincare	Mini Dramatically Different Moisturizing Gel	4.5	3000	14700	15.0	15.0	False	no options	0	16.913043
1661	St. Tropez	Skincare	Self Tan Express Bronzing Mousse Mini	4.5	10	3600	19.0	19.0	False	no options	0	16.913043
5027	Farmacy	Skincare	Mini Green Clean Makeup Meltaway Cleansing Balm	4.5	78	0	22.0	22.0	True	no options	1	16.913043

Rating tertinggi dari category perfume

```
[ ] perfume = pd.DataFrame(df_sephora.loc[df_sephora.category == 'Perfume', :].sort_values(by='rating', ascending=False))
perfume.head()
```

	brand	category	name	rating	number_of_reviews	love	price	value_price	MarketingFlags	options	exclusive
1700	Versace	Perfume	Bright Crystal & Dylan Blue Pour Femme Ovetto ...	5.0	61	12100	30.0	30.0	True	- 0.34 oz/ 10 mL Eau de Toilette Spray	1
4809	CLINIQUE	Perfume	Clinique Happy Heart	5.0	4	1400	53.0	53.0	True	- 1 oz/ 30 mL Eau de Parfum Spray - 1.7 oz/ 5...	0
825	TOM FORD	Perfume	Champaca Absolute	5.0	18	2400	240.0	240.0	False	- 1.7 oz/ 50 mL Eau de Parfum	0
816	Atelier Cologne	Perfume	Oolang Infini Cologne Absolue Pure Perfume	5.0	4	2000	80.0	80.0	True	- Out of stock: 1 oz/ 30 mL Cologne Absolue P...	1
7176	TOM FORD	Perfume	Ombre Leather All Over Body Spray	5.0	2	701	49.0	49.0	False	no options	0

```
[ ] ratahargaperfume = perfume["price"].mean()
ratahargaperfume
```

109.52580645161291

Rata-rata harga perfume

```
[ ] perfume['rata - rata harga'] = ratahargaperfume
perfume.head(5)
```

	brand	category	name	rating	number_of_reviews	love	price	value_price	MarketingFlags	options	exclusive	rata - rata harga
1700	Versace	Perfume	Bright Crystal & Dylan Blue Pour Femme Ovetto ...	5.0	61	12100	30.0	30.0	True	- 0.34 oz/ 10 mL Eau de Toilette Spray	1	109.525806
4809	CLINIQUE	Perfume	Clinique Happy Heart	5.0	4	1400	53.0	53.0	True	- 1 oz/ 30 mL Eau de Parfum Spray - 1.7 oz/ 5...	0	109.525806
825	TOM FORD	Perfume	Champaca Absolute	5.0	18	2400	240.0	240.0	False	- 1.7 oz/ 50 mL Eau de Parfum	0	109.525806
816	Atelier Cologne	Perfume	Oolang Infini Cologne Absolue Pure Perfume	5.0	4	2000	80.0	80.0	True	- Out of stock: 1 oz/ 30 mL Cologne Absolue P...	1	109.525806
7176	TOM FORD	Perfume	Ombre Leather All Over Body Spray	5.0	2	701	49.0	49.0	False	no options	0	109.525806

<https://github.com/elsbt/TechnicalTest.git>