In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

import libraries for data manipulation and visualization

In [3]:
```
data = pd.read_csv('data.csv')
```

read in data

In [4]:
```
data.head()
```

Out[4]:

	STUDYID	RPT	time	death	BMI	HEIGHTBL	WEIGHTBL	ALP	ALT	AST
0	ASC	ASC-001-0003	585	1	28.7000	172.72	85.73000	5.733341	2.944439	3.258097
1	ASC	ASC-001-0004	495	1	25.3000	171.80	74.80000	6.269096	2.708050	3.295837
2	ASC	ASC-001-0005	167	1	26.2000	167.90	73.90000	5.068904	2.639057	2.890372
3	ASC	ASC-001-0008	161	1	26.9915	166.70	83.50879	4.812184	3.761200	3.688879
4	ASC	ASC-001-0009	575	0	46.5000	175.60	143.30000	3.496508	2.944439	3.258097

5 rows × 105 columns

Cut off from screen

In [18]:
```
list(data.columns)
```

Out[18]:
```
['STUDYID',
 'RPT',
 'time',
 'death',
 'BMI',
 'HEIGHTBL',
 'WEIGHTBL',
 'ALP',
 'ALT',
 'AST',
 'CA',
 'CREAT',
 'HB',
 'LDH',
 'NEU',
 'PLT',
 'PSA',
 'TBILI',
 'TESTO',
 'WBC',
 'CREACL',
```

Identifiers → Not useful in Data Analysis

Target data → See how other variables affect these 2

body measurements → BMI/Weight could make a difference

Baseline Lab Values → Definitely markers for disease

```
'NA.',
'MG',
'PHOS',
'ALB',
'TPRO',
'RBC',
'LYM',
'BUN',
'CCRC',
'GLU',
'SYSTOLICBP',
'DIASTOLICBP',
'PULSE',
'HEMAT',
'SPEGRA',
'LYMperLEU',
'MONO',
'MONOperLEU',
'NEUperLEU',
'POT',
'BASOperLEU',
'EOS',
'EOSperLEU',
'TARGET',
'LYMPH_NODES',
'KIDNEYS',
'LUNGS',
'LIVER',
'PLEURA',
'OTHER',
'PROSTATE',
'ORCHIDECTOMY',
'PROSTATECTOMY',
'LYMPHADENECTOMY',
'BILATERAL_ORCHIDECTOMY',
'PRIOR_RADIOTHERAPY',
'ANALGESICS',
'ANTI_ANDROGENS',
'GLUCOCORTICOID',
'GONADOTROPIN',
'BISPHOSPHONATE',
'CORTICOSTEROID',
'IMIDAZOLE',
'ACE_INHIBITORS',
'BETA_BLOCKING',
'HMG_COA_REDUCT',
'ESTROGENS',
'ANTI_ESTROGENS',
'CEREBACC',
'CHF',
'DVT',
'DIAB',
'MI',
'PULMEMB',
'SPINCOMP',
'COPD',
'MHBLOOD',
'MHCARD',
'MHCONGEN',
'MHEAR',
'MHENDO',
'MHGASTRO',
'MHHEPATO',
'MHIMMUNE',
'MHINFECT',
```

??? Not in Data Dictionary

Blood Pressure
Pulse

Not sure
- would skip for now

Lesions

Not sure what this is

- Mets

Treatments (procedures + Medicines)

Procedures

- Medicines

Medical History (Diseases)

- Cerebrovascular accident - Stroke
- Congestive Heart Failure
- Deep Venous Thrombosis - Blood Clot in Deep Vein (Leg)
- Diabetes
- Myocardial Infarction - Heart Attack
- Pulmonary Embolism - Blood Clot in Artery in Lung
- Spinal Cord Compression
- Chronic Obstructive Pulmonary Disease

Medical History (Body Systems)

Y/N for any disease in that system

```
'MHINJURY',
'MHINVEST',
'MHMETAB',
'MHPSYCH',
'MHRENAL',
'MHRESP',
'MHSKIN',
'MHVASC',
'ECOG_C',
'AGEGRP2',
'RaceAsian',
'RaceBlack',
'RaceOther',
'RaceWhite',
'RegionAsia',
'RegionEastEuro',
'RegionNorthAmer',
'RegionSouthAmer',
'RegionWestEuro']
```

BL Patient Performance Status → BL Patient Status: 0 = Fully active; 1 = Restricted activity; 2 = No activity; 3 = limited self care; 4 = bed-bound — not in dataset

Age Group → Age Group: 0 = 18-64; 1 = 65-74; 2 = 75+

Race

Region

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1600 entries, 0 to 1599
Columns: 105 entries, STUDYID to RegionWestEuro
dtypes: float64(40), int64(63), object(2)
memory usage: 1.3+ MB
```

In [12]:
```
data.describe()
```

Out[12]:

	time	death	BMI	HEIGHTBL	WEIGHTBL	ALP	
count	1600.000000	1600.000000	1600.000000	1600.000000	1600.000000	1600.000000	1600.00
mean	453.798750	0.414375	28.204521	174.089024	85.506196	5.041073	3.008
std	289.693734	0.492768	4.531143	7.713103	15.834734	0.863298	0.509
min	4.000000	0.000000	15.900000	131.500000	46.000000	3.295837	1.386
25%	244.750000	0.000000	25.175000	169.200000	75.000000	4.406719	2.708
50%	387.000000	0.000000	27.700000	174.400000	84.000000	4.828314	2.95
75%	581.000000	1.000000	30.700000	179.500000	93.400000	5.494088	3.29
max	1594.000000	1.000000	54.300000	198.700000	164.700000	8.289791	5.18

avg survival time of ~1 year

~4.4 years

8 rows × 103 columns

In [7]:
```
data.corr()['time'].sort_values(ascending=True)
```

Pearson Correlation Coefficient between each variable, sorted

Out[7]:
```
TESTO                -0.207605
ALP                  -0.152988
LDH                  -0.141076
NEUperLEU            -0.134277
GLU                  -0.117638
                        ...
TBILI                 0.160275
RegionSouthAmer       0.173898
CORTICOSTEROID        0.205400
```

high lab values correlate with shorter survival time

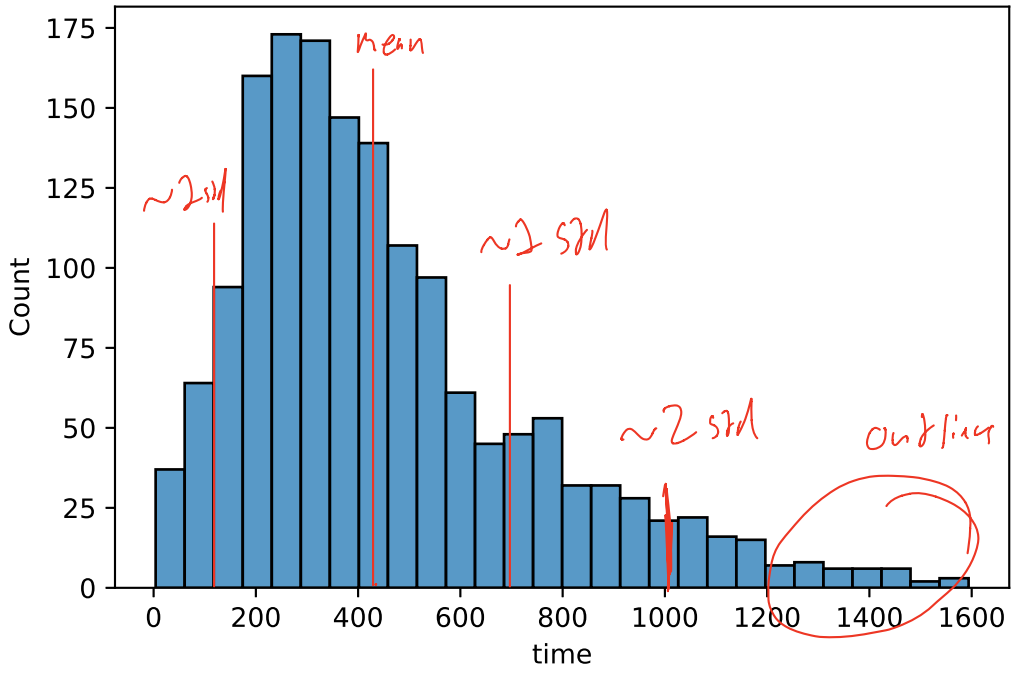
corticosteroid use correlates with longer survival time

```
time                1.000000
RegionAsia               NaN
Name: time, Length: 103, dtype: float64
```

In [11]:
```
sns.histplot(data['time'])
```

Normal distribution of survival time

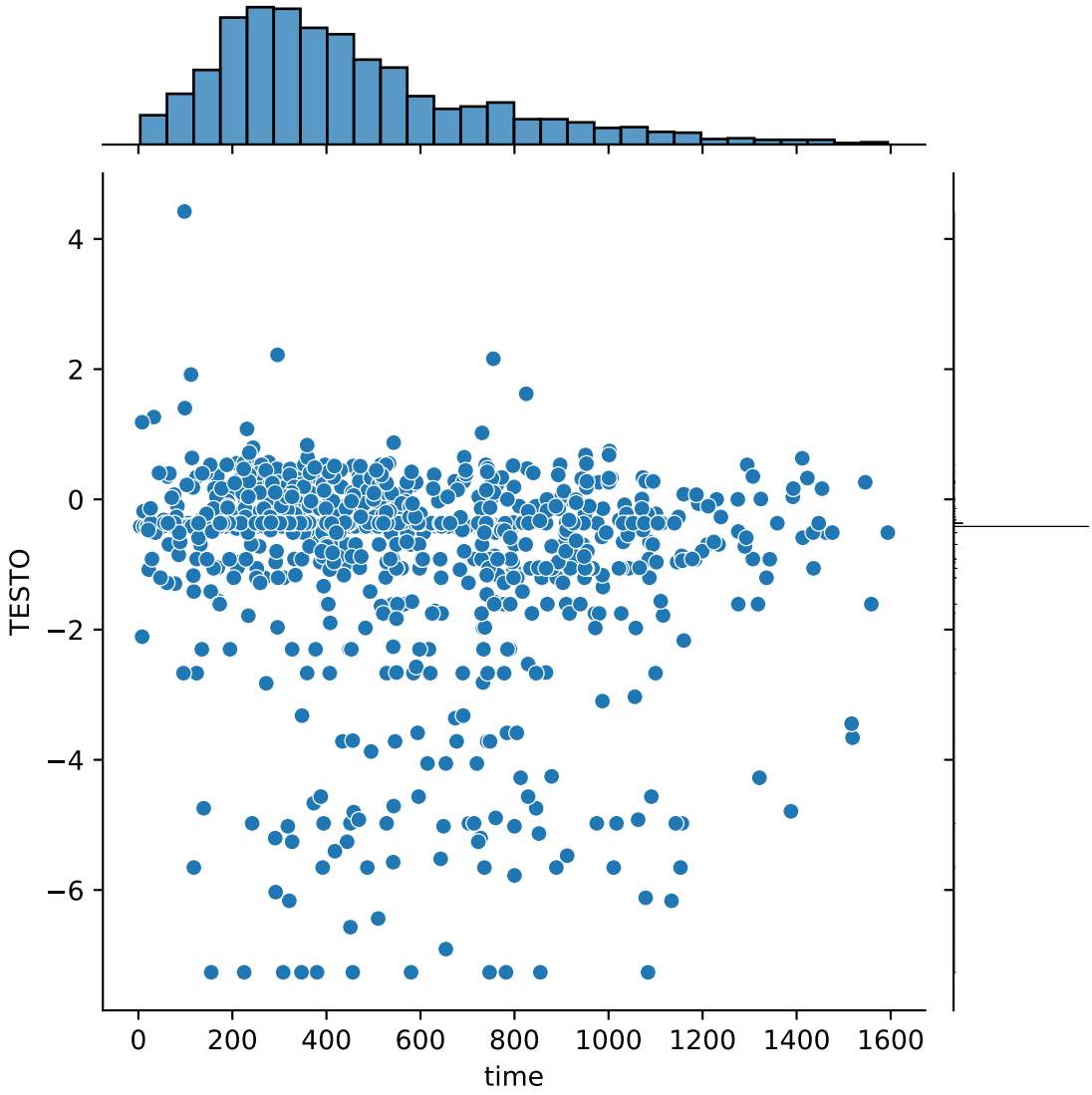
Out[11]: <AxesSubplot:xlabel='time', ylabel='Count'>



In [19]:
```
sns.jointplot(x='time',y='TESTO', data=data)
```

Out[19]: <seaborn.axisgrid.JointGrid at 0x7f90885b2af0>

Compare survival time with testosterone level



In [34]:
```
corr_data = data.corr()['time'].sort_values(ascending=False)
```

In [39]:
```
corr_data.head()
```

Out[39]:
```
time                1.000000
CORTICOSTEROID      0.205400
RegionSouthAmer     0.173898
TBILI               0.160275
HB                  0.149179
Name: time, dtype: float64
```

- CORTICOSTEROID — corticosteroid (Medicine)
- RegionSouthAmer — Being in South America (Region)
- TBILI — Total Bilirubin (Liver Lab)
- HB — Hemoglobin (Blood Lab)

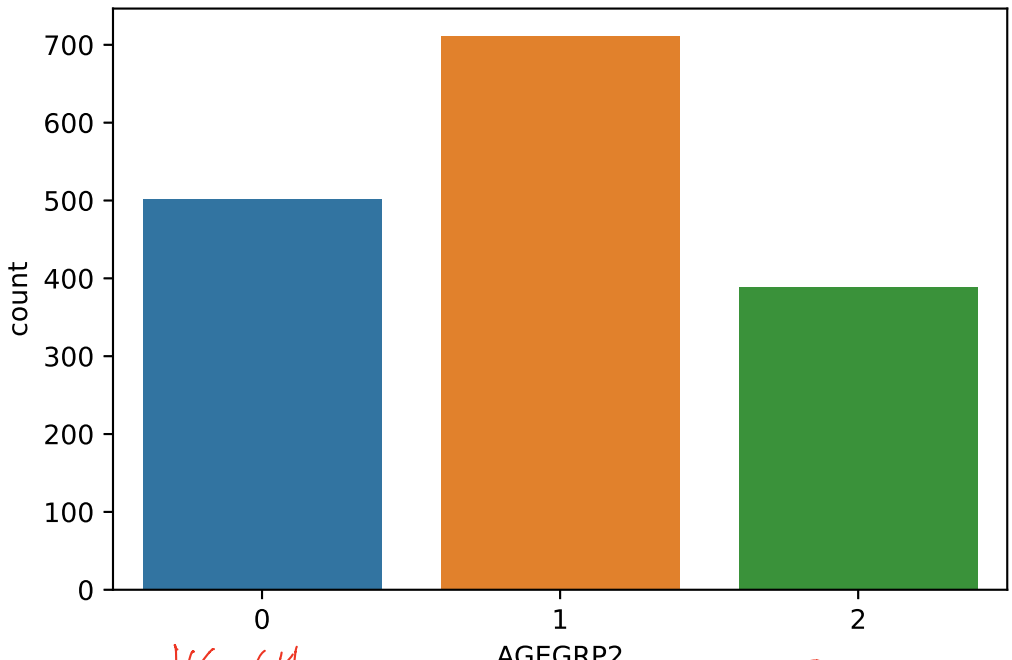
In [44]:
```
data['AGEGRP2']
```

Out[44]:
```
0       1
1       2
2       1
3       2
4       1
       ..
1595    0
1596    1
1597    0
```

```
1598    0
1599    1
Name: AGEGRP2, Length: 1600, dtype: int64
```

In [172…

```
sns.countplot(x=data['AGEGRP2']).set_axis_labels = ('18-64','65-74','75+')
```



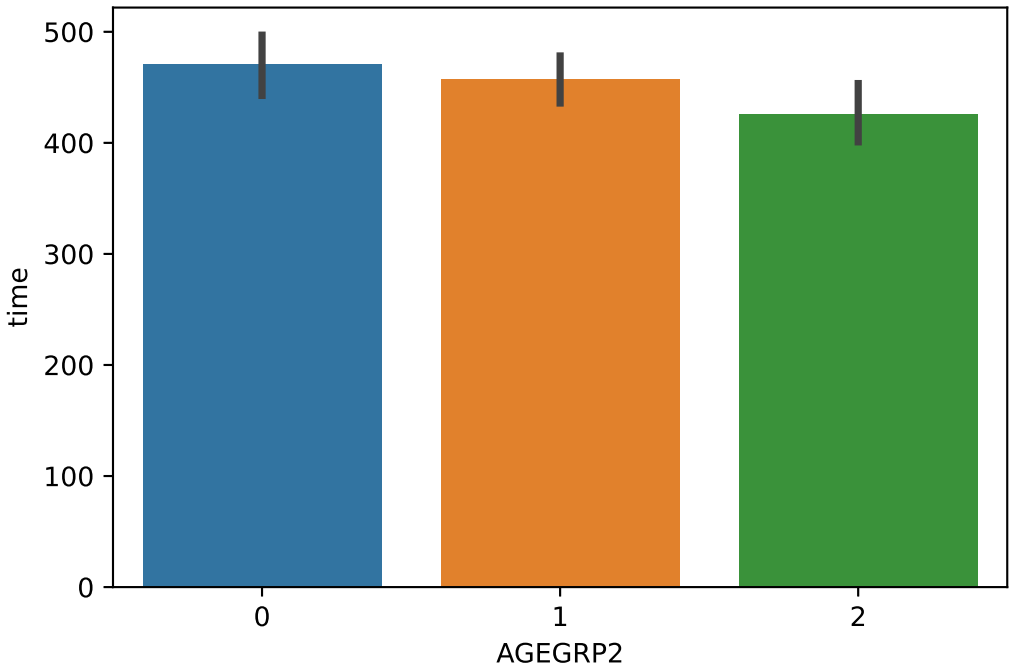
kind of even split

18-64 65-74 75+

In [162…

```
sns.barplot(x='AGEGRP2',y='time',data=data)
```

Out[162… <AxesSubplot:xlabel='AGEGRP2', ylabel='time'>



age doesn't have big effect on survival time

In [71]:

```
data['BMI']
```

```
0       28.7000
```

Out[71]:
```
1       25.3000
2       26.2000
3       26.9915
4       46.5000
          ...
1595    32.4000
1596    31.0000
1597    26.0000
1598    30.6000
1599    30.4000
Name: BMI, Length: 1600, dtype: float64
```

In [77]:
```
data.corr()['death'].sort_values(ascending=True)
```

Pearson coefficient — related to death

Out[77]:
```
ALB              -0.238652
TESTO            -0.180615
HEMAT            -0.177278
HB               -0.153038
RBC              -0.142311
                    ...
LDH               0.206975
CORTICOSTEROID    0.211477
ALP               0.227153
death             1.000000
RegionAsia             NaN
Name: death, Length: 103, dtype: float64
```

same as time

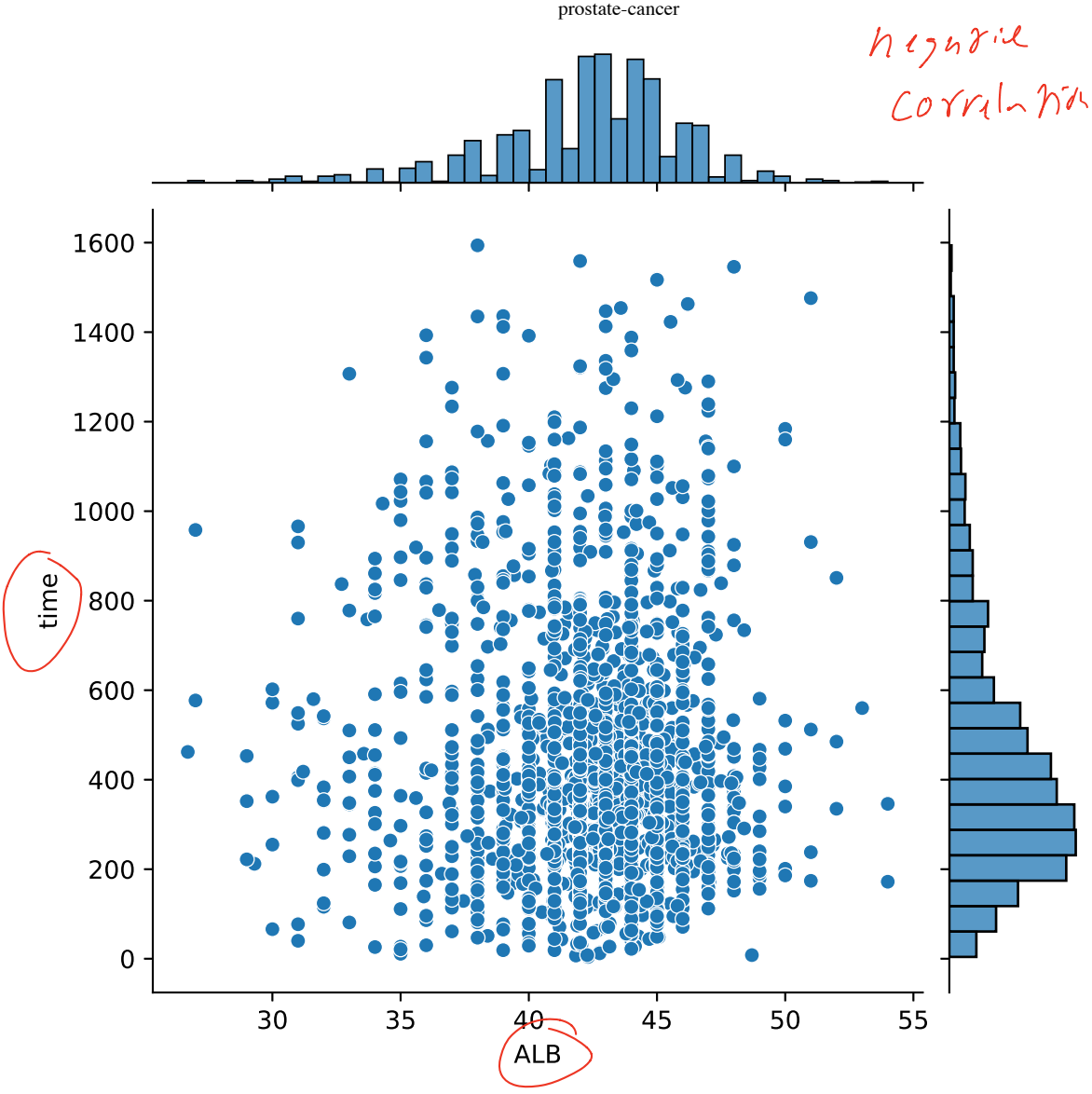
opposite as time

same

opp

In [91]:
```
sns.jointplot(x='ALB',y='time', data=data)
```

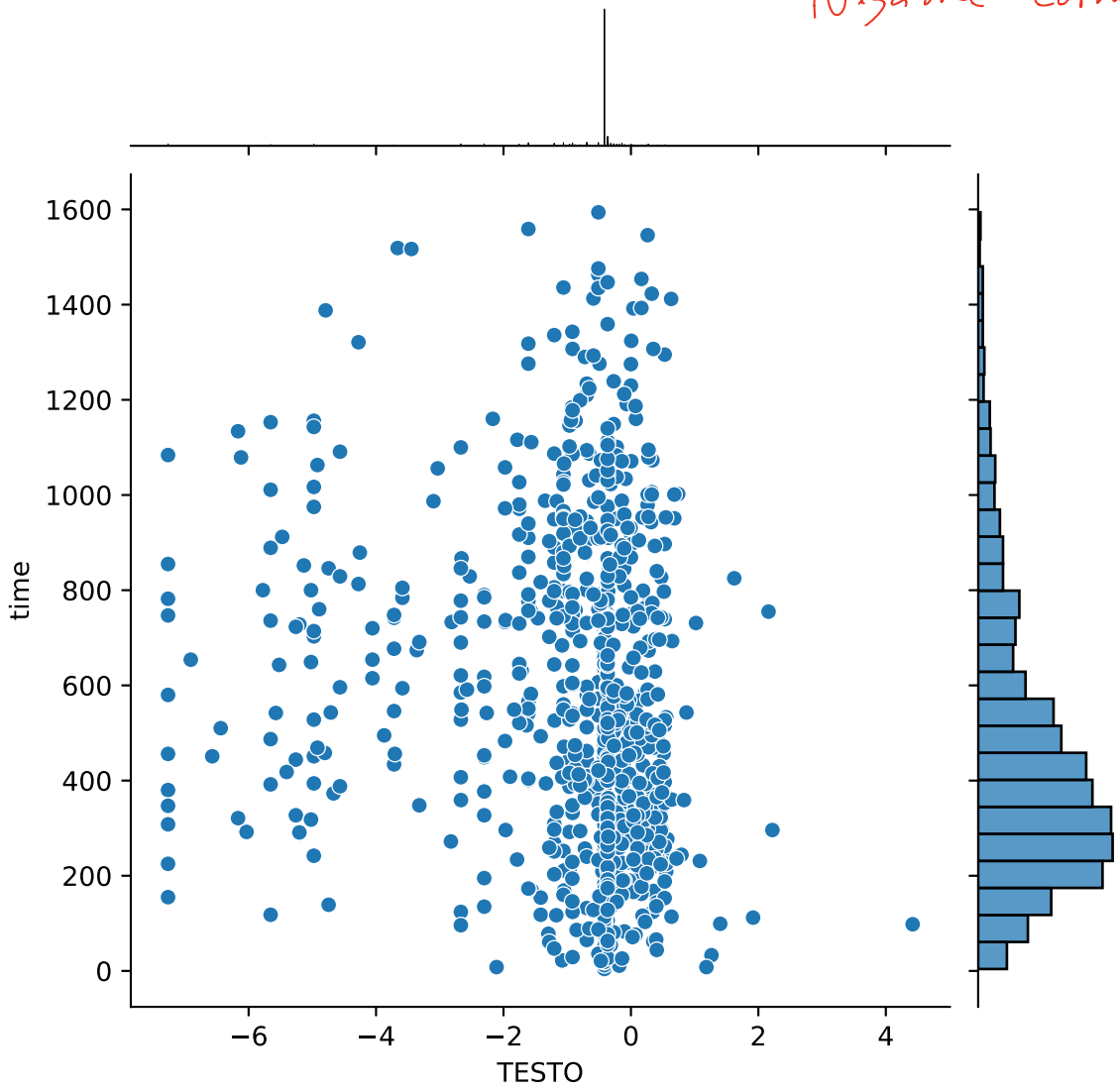
Out[91]:
```
<seaborn.axisgrid.JointGrid at 0x7f9071eb0280>
```



In [92]:
```
sns.jointplot(x='TESTO',y='time', data=data)
```

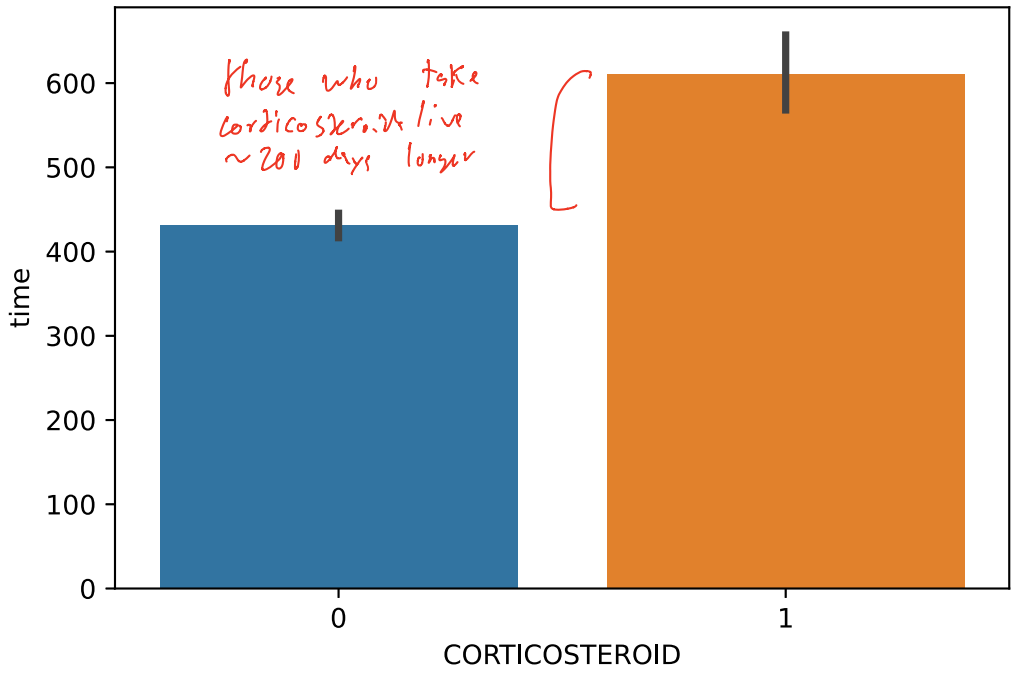
Out[92]:
```
<seaborn.axisgrid.JointGrid at 0x7f9072179af0>
```

Negative correlation



In [94]:
```
sns.barplot(x='CORTICOSTEROID',y='time', data=data)
```

Out[94]:
```
<AxesSubplot:xlabel='CORTICOSTEROID', ylabel='time'>
```



Treatments

In [96]:
```
treatments =  ['ORCHIDECTOMY','PROSTATECTOMY','LYMPHADENECTOMY','BILATERAL_ORCHI
```

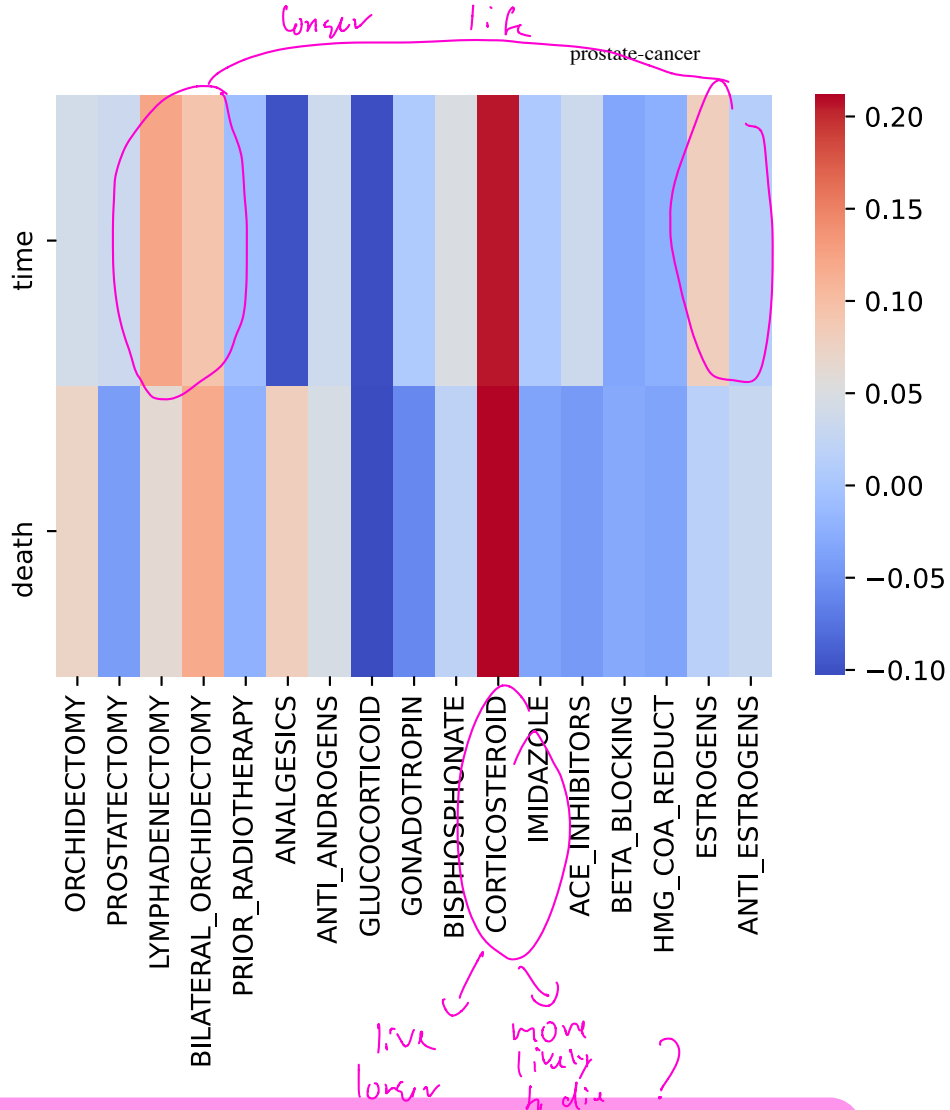
In [111…
```
data.corr()[treatments].loc[['time', 'death']]
```

Out[111…

	ORCHIDECTOMY	PROSTATECTOMY	LYMPHADENECTOMY	BILATERAL_ORCHIDECTOMY	PR
time	0.041008	0.033968	0.12393	0.094407	
death	0.073053	-0.041368	0.06353	0.118714	

In [122…
```
sns.heatmap(data.corr()[treatments].loc[['time', 'death']], cmap="coolwarm")
```

Out[122… <AxesSubplot:>



Corticosteroids seem to have greatest impact on patient outcome

Lab Values

In [128…

```
lab_values = [
 'ALP',
 'ALT',
 'AST',
 'CA',
 'CREAT',
 'HB',
 'LDH',
 'NEU',
 'PLT',
 'PSA',
 'TBILI',
 'TESTO',
 'WBC',
 'CREACL',
 'NA.',
 'MG',
 'PHOS',
 'ALB',
 'TPRO',
 'RBC',
 'LYM',
 'BUN',
```

```
 'CCRC',
 'GLU']
```

In [130…

```
data.corr()[lab_values].loc[['time', 'death']]
```

Out[130…

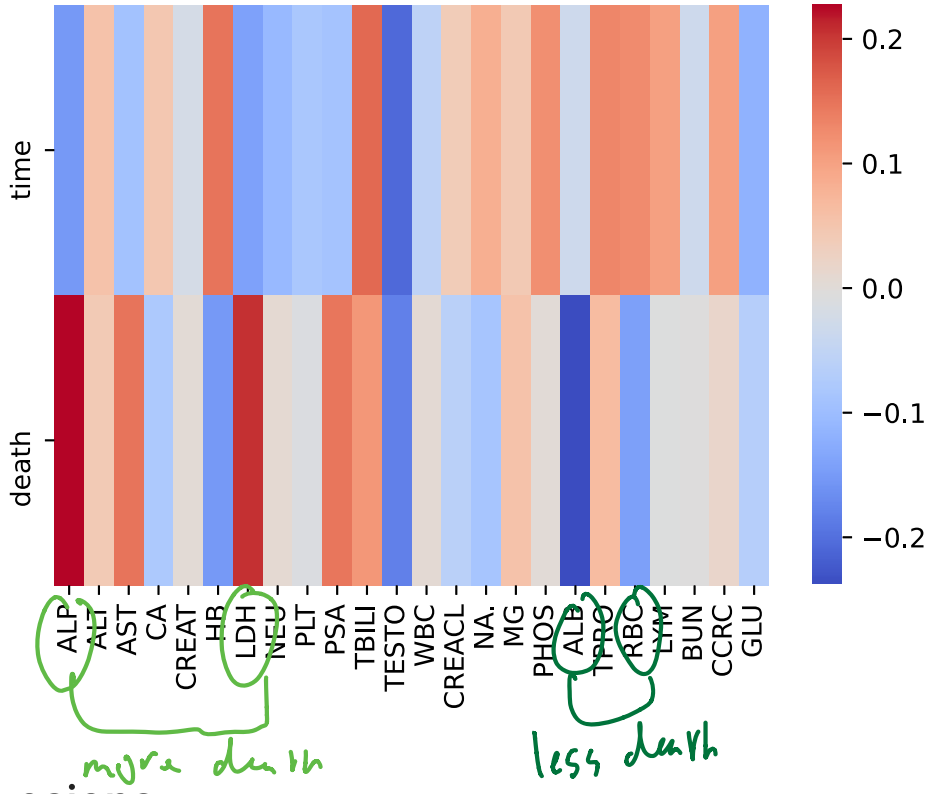
	ALP	ALT	AST	CA	CREAT	HB	LDH	NEU	
time	-0.152988	0.055176	-0.092459	0.048479	-0.021601	0.149179	-0.141076	-0.107470	-0.(
death	0.227153	0.041613	0.149438	-0.079346	0.003589	-0.153038	0.206975	0.008491	-0.(

2 rows × 24 columns

In [131…

```
sns.heatmap(data.corr()[lab_values].loc[['time', 'death']], cmap="coolwarm")
```

Out[131… <AxesSubplot:>



Lesions

In [132…

```
lesions = [
 'TARGET',
 'LYMPH_NODES',
 'KIDNEYS',
 'LUNGS',
 'LIVER',
 'PLEURA',
 'OTHER',
 'PROSTATE'
]
```

In [134…

```
data.corr()[lesions].loc[['time', 'death']]
```

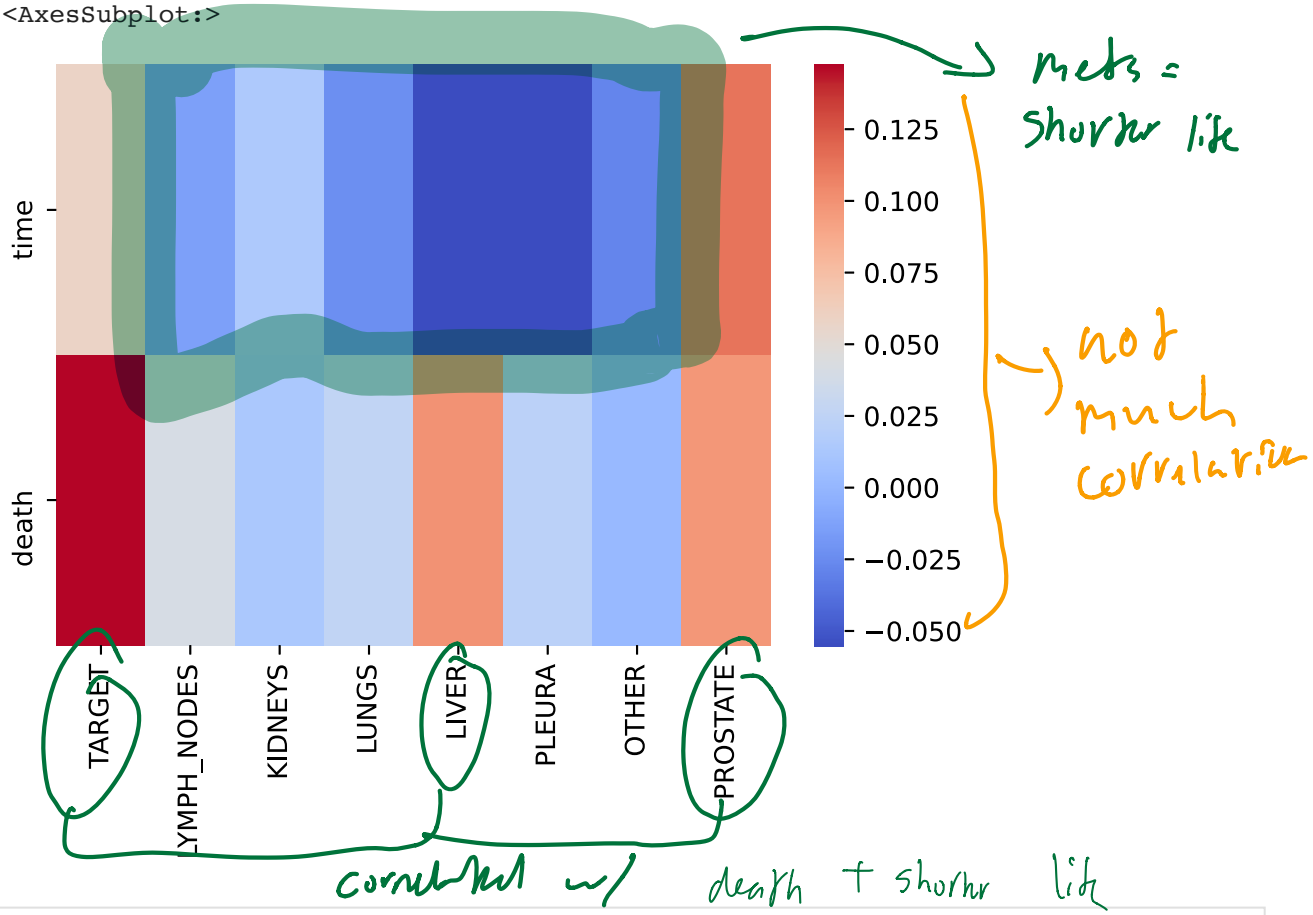
Out[134…

	TARGET	LYMPH_NODES	KIDNEYS	LUNGS	LIVER	PLEURA	OTHER	PROSTA
time	0.058253	-0.013224	0.014979	-0.023299	-0.054362	-0.054945	-0.027685	0.1140
death	0.147467	0.040938	0.013785	0.026993	0.100168	0.023198	0.002114	0.0982

In [136…

```
sns.heatmap(data.corr()[lesions].loc[['time', 'death']], cmap="coolwarm")
```

Out[136… <AxesSubplot:>



In [151…

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

Out[151…

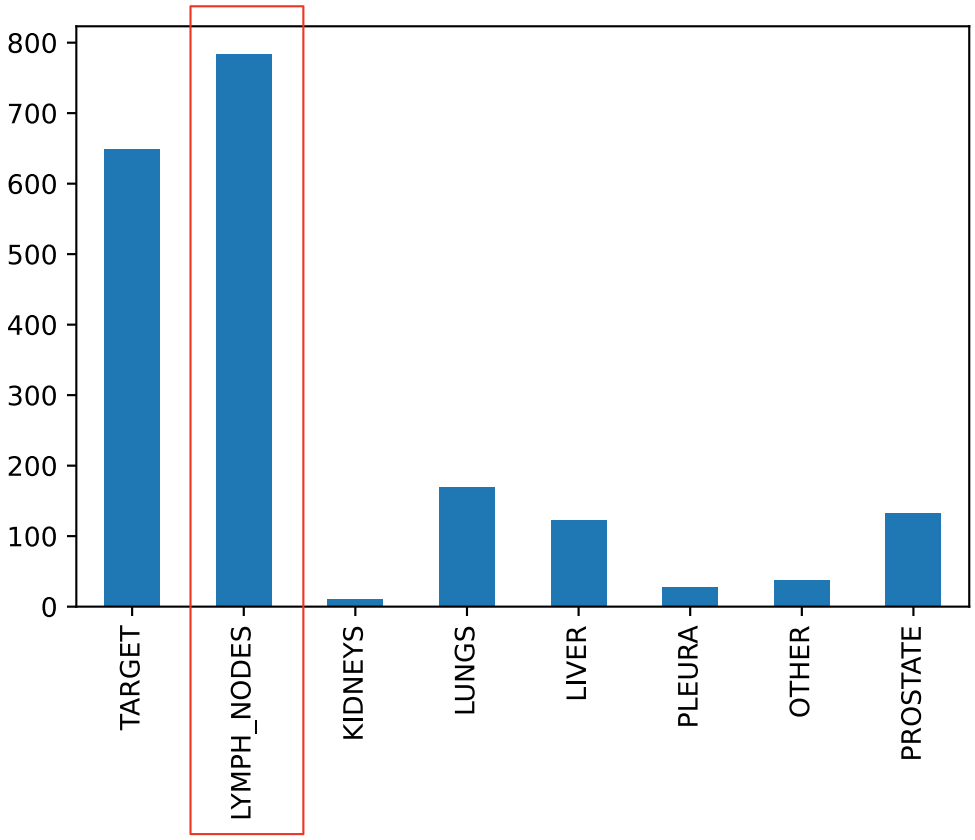
```
TARGET          649
LYMPH_NODES     784
KIDNEYS          10
LUNGS           170
LIVER           123
PLEURA           28
OTHER            38
PROSTATE        132
dtype: int64
```

In [160…

```
data[lesions].sum().plot(kind='bar')
```

Out[160… <AxesSubplot:>

most mets in lymph nodes



BMI - Height/Weight

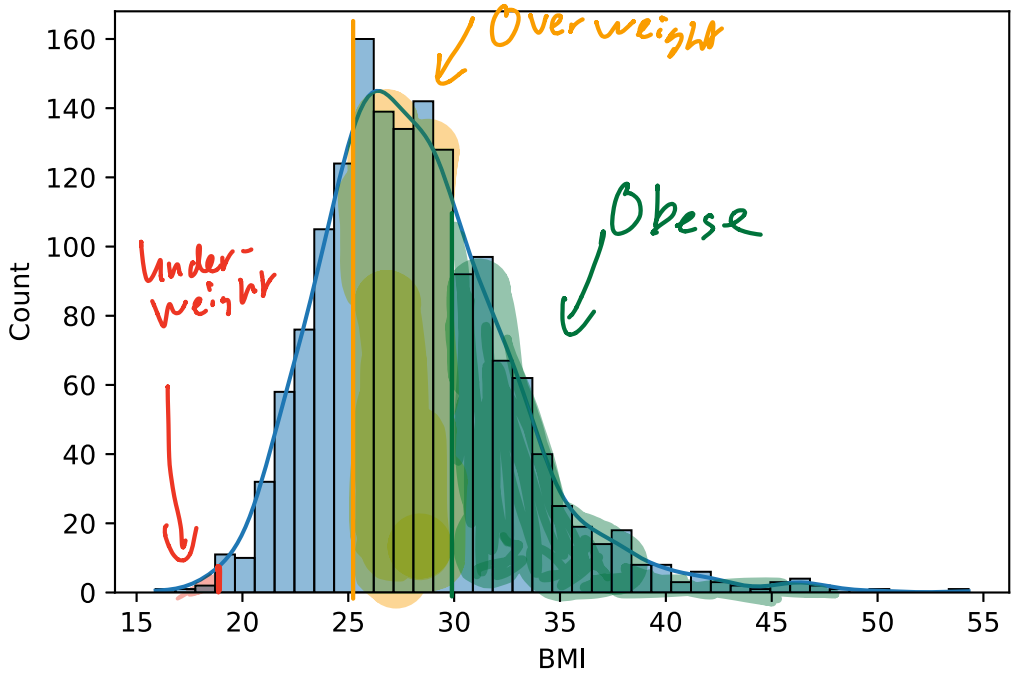
In [178…

```
body_measures = [
 'BMI',
 'HEIGHTBL',
 'WEIGHTBL'
]
```

In [189…

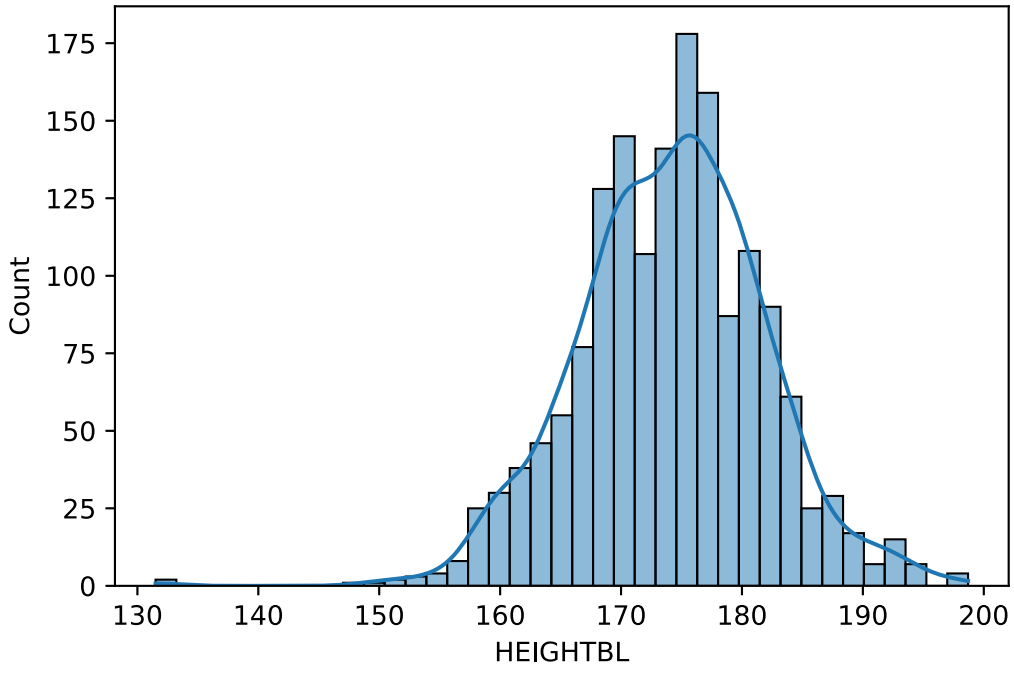
```
sns.histplot(data['BMI'], kde=True)
```

Out[189… <AxesSubplot:xlabel='BMI', ylabel='Count'>



In [190…
```
sns.histplot(data['HEIGHTBL'], kde=True)
```

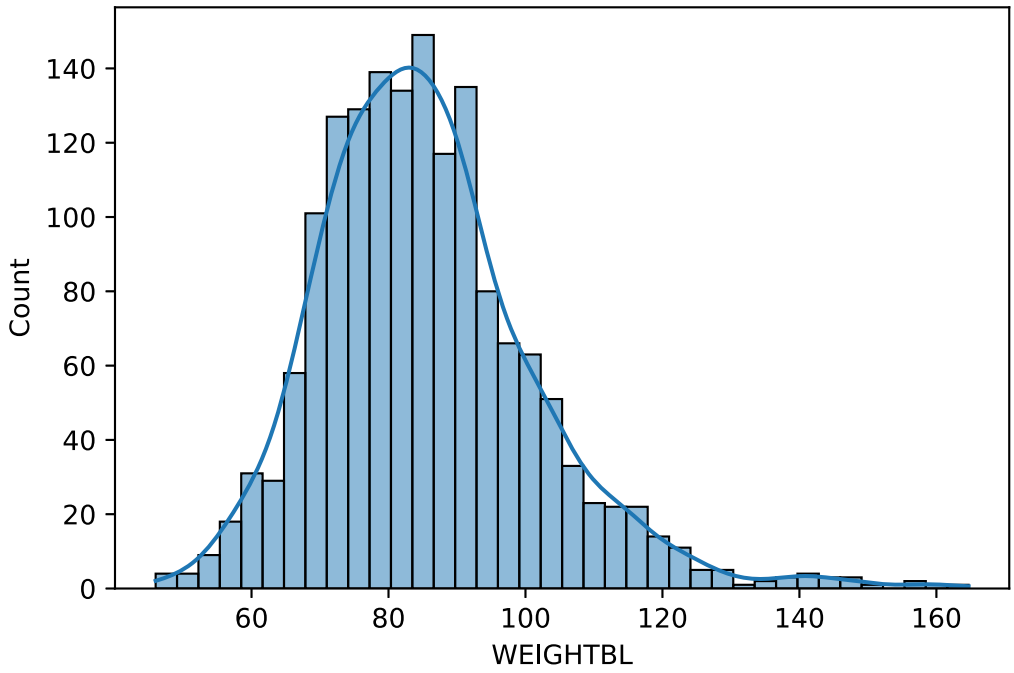
Out[190… <AxesSubplot:xlabel='HEIGHTBL', ylabel='Count'>



Useful for BMI, not alone

In [191…
```
sns.histplot(data['WEIGHTBL'], kde=True)
```

Out[191… <AxesSubplot:xlabel='WEIGHTBL', ylabel='Count'>



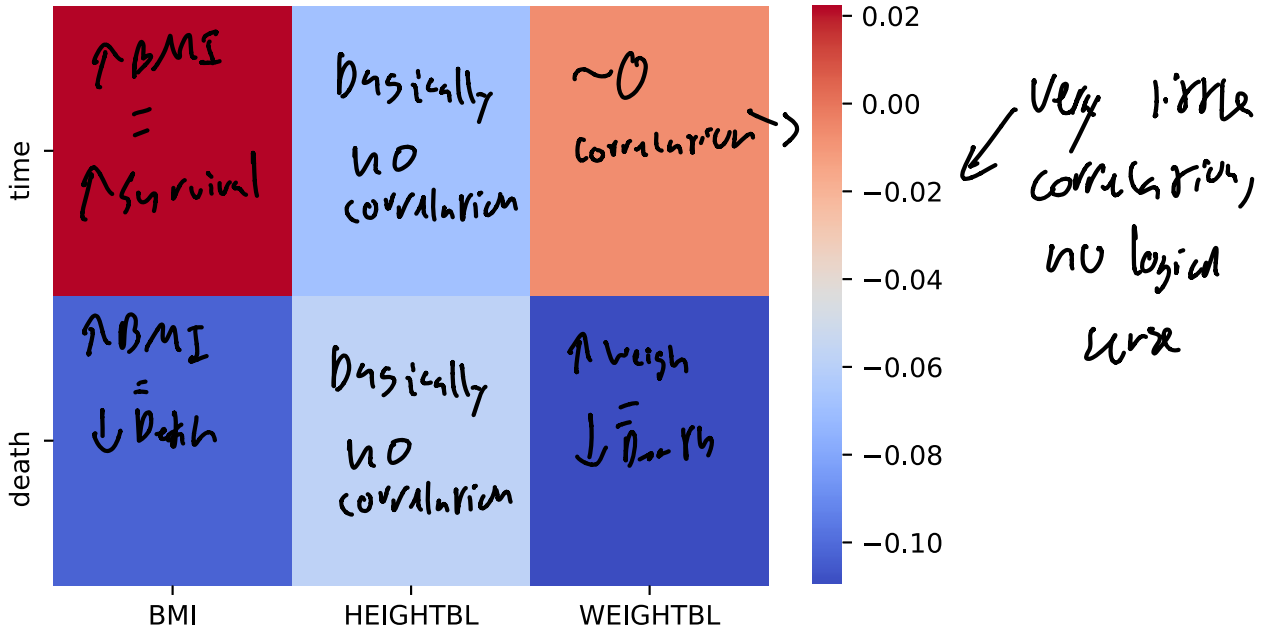
In [195…
```
data.corr()[body_measures].loc[['time','death']]
```

Out[195…

	BMI	HEIGHTBL	WEIGHTBL
time	0.022269	-0.068792	-0.006945
death	-0.102690	-0.057813	-0.109833

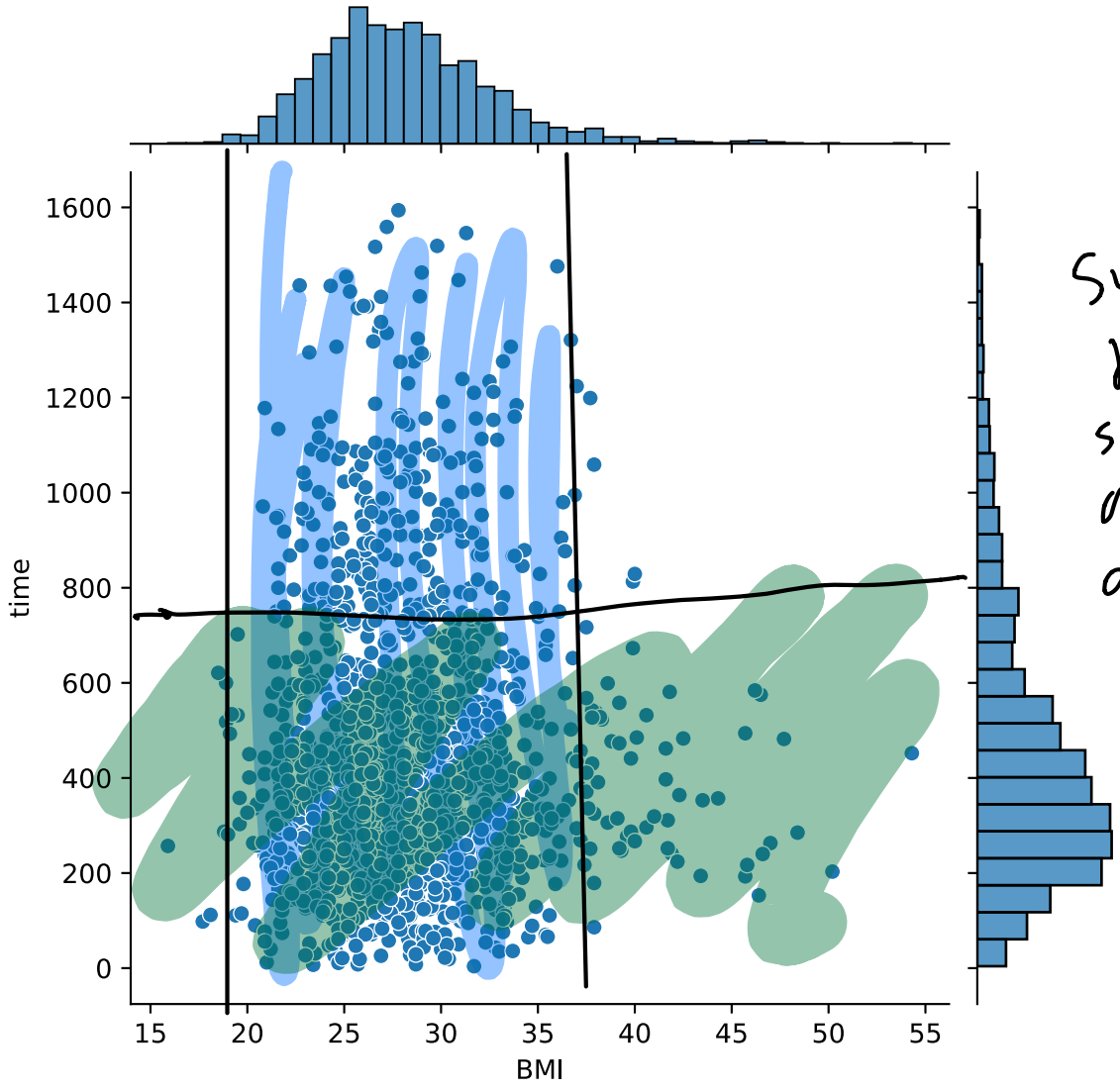
In [196…
```
sns.heatmap(data.corr()[body_measures].loc[['time', 'death']], cmap="coolwarm")
```

Out[196… <AxesSubplot:>



In [197…
```
sns.jointplot(x='BMI',y='time',data=data)
```

Out[197… <seaborn.axisgrid.JointGrid at 0x7f907df85700>

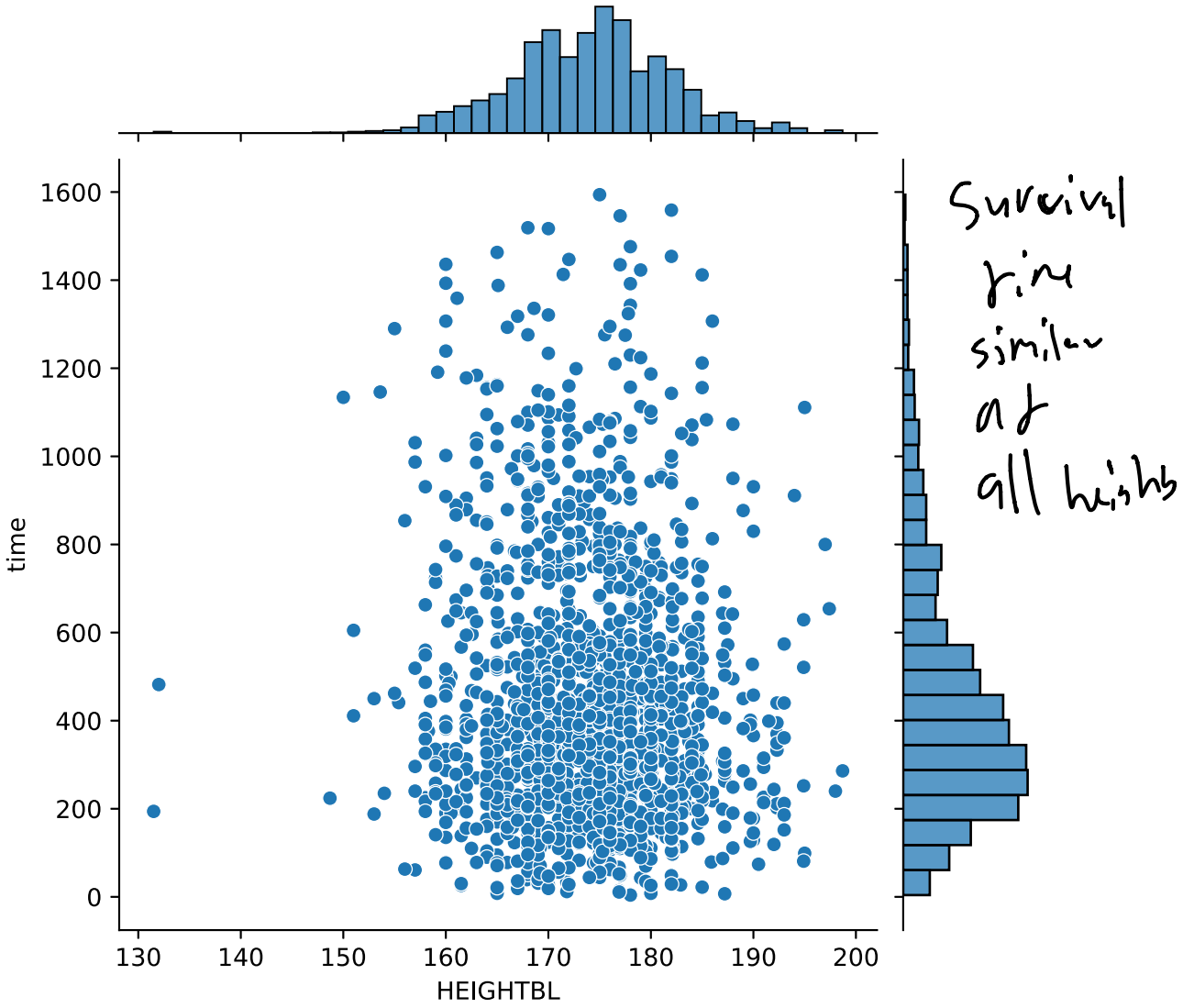


Survival going similar at all BMIs

In [198…

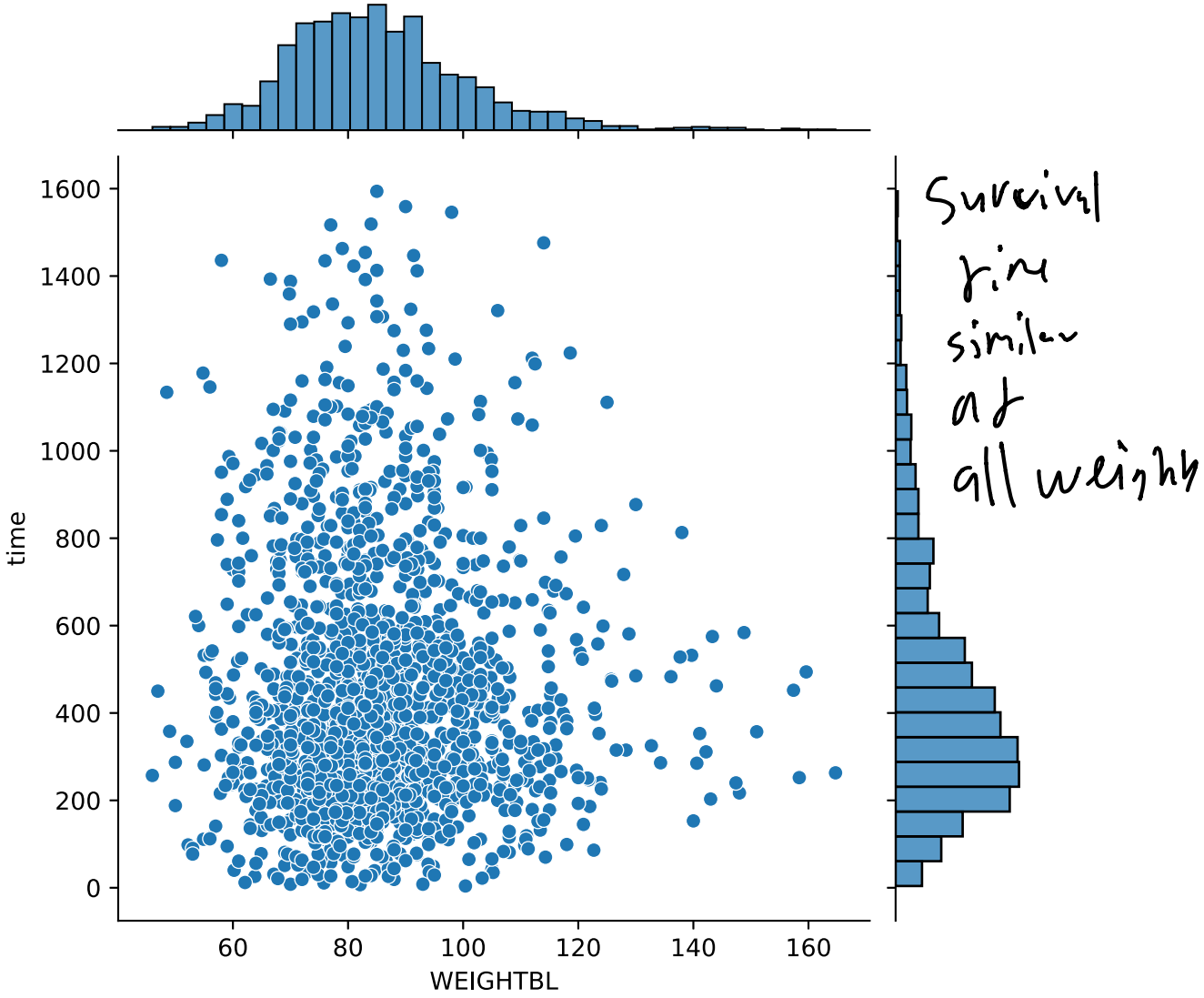
```
sns.jointplot(x='HEIGHTBL',y='time',data=data)
```

Out[198… <seaborn.axisgrid.JointGrid at 0x7f907e34ccd0>



In [199…
```
sns.jointplot(x='WEIGHTBL',y='time',data=data)
```

Out[199… <seaborn.axisgrid.JointGrid at 0x7f907e55b3a0>



Medical History — Diseases

In [200…

```
mhDisease =  ['CEREBACC',
 'CHF',
 'DVT',
 'DIAB',
'MI',
 'PULMEMB',
 'SPINCOMP',
 'COPD']
```

In [202…

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

Out[202…

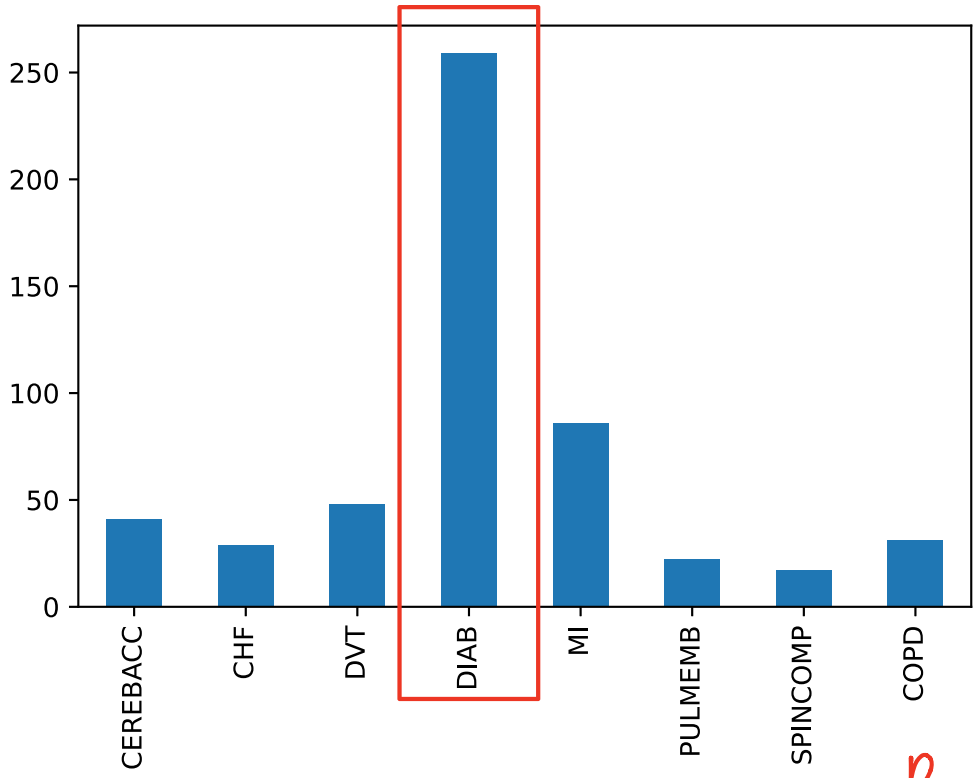
```
CEREBACC      41
CHF           29
DVT           48
DIAB         259
MI            86
PULMEMB       22
SPINCOMP      17
```

→ most common disease

```
COPD          31
dtype: int64
```

In [203…
```
data[mhDisease].sum().plot(kind='bar')
```

Out[203… <AxesSubplot:>



Pearson Correlation

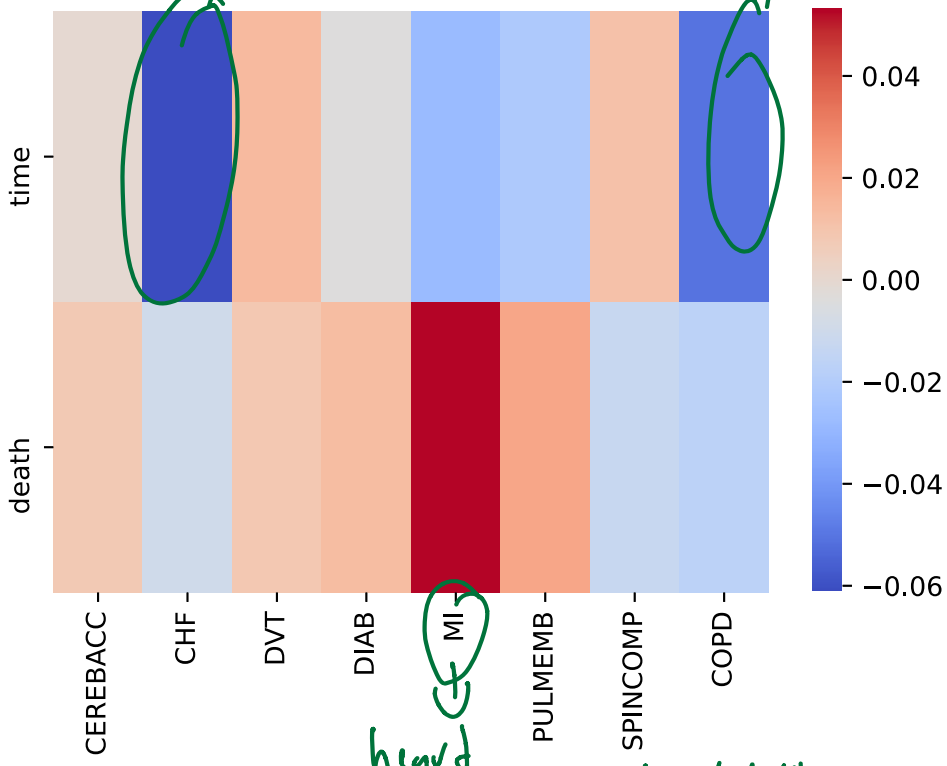
In [205…
```
data.corr()[mhDisease].loc[['time','death']]
```

Out[205…

	CEREBACC	CHF	DVT	DIAB	MI	PULMEMB	SPINCOMP	COPD
time	-0.000611	-0.061331	0.013545	-0.004294	-0.028801	-0.022101	0.011144	-0.050834
death	0.008115	-0.009671	0.008256	0.012665	0.052678	0.020524	-0.012924	-0.016988

In [207…
```
sns.heatmap(data.corr()[mhDisease].loc[['time','death']], cmap='coolwarm')
```

Out[207… <AxesSubplot:>



? Heart failure + Pulmonary disease = shorter life

Generally, other diseases = lower survival time + higher chance of death, not that accurate because of little data w/ other diseases

heart attack correlated w/ death

Medical History - Body System

In [217…

```
mhBody = ['MHBLOOD',
 'MHCARD',
 'MHCONGEN',
 'MHEAR',
 'MHENDO',
 'MHGASTRO',
 'MHHEPATO',
 'MHIMMUNE',
 'MHINFECT',
 'MHINJURY',
 'MHINVEST',
 'MHMETAB',
 'MHPSYCH',
 'MHRENAL',
 'MHRESP',
 'MHSKIN',
 'MHVASC']
```

In [219…

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

Out[219…

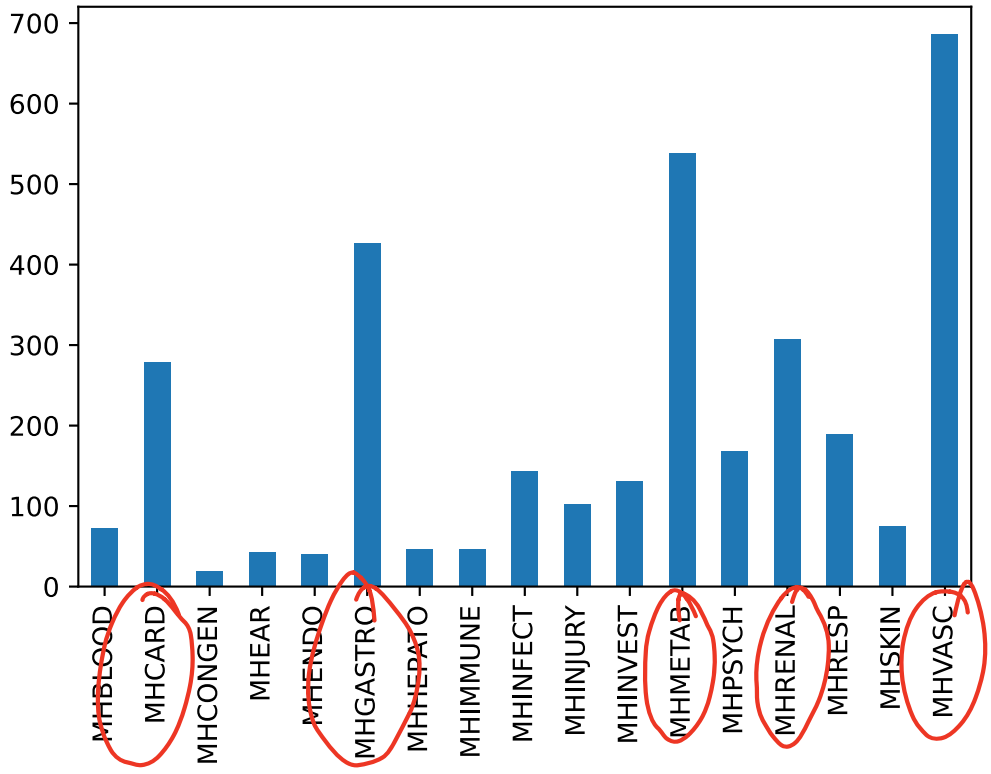
```
MHBLOOD      73
MHCARD      279
MHCONGEN     19
MHEAR        43
MHENDO       40
MHGASTRO    427
MHHEPATO     46
MHIMMUNE     46
MHINFECT    144
MHINJURY    103
```

not even spread

```
MHINVEST     131
MHMETAB      539
MHPSYCH      168
MHRENAL      307
MHRESP       190
MHSKIN        75
MHVASC       686
dtype: int64
```

In [218…
```
data[mhBody].sum().plot(kind='bar')
```

Out[218… <AxesSubplot:>



1. Vascular
2. Metabolic
3. Gastro-intestinal
4. Renal
5. Cardiac

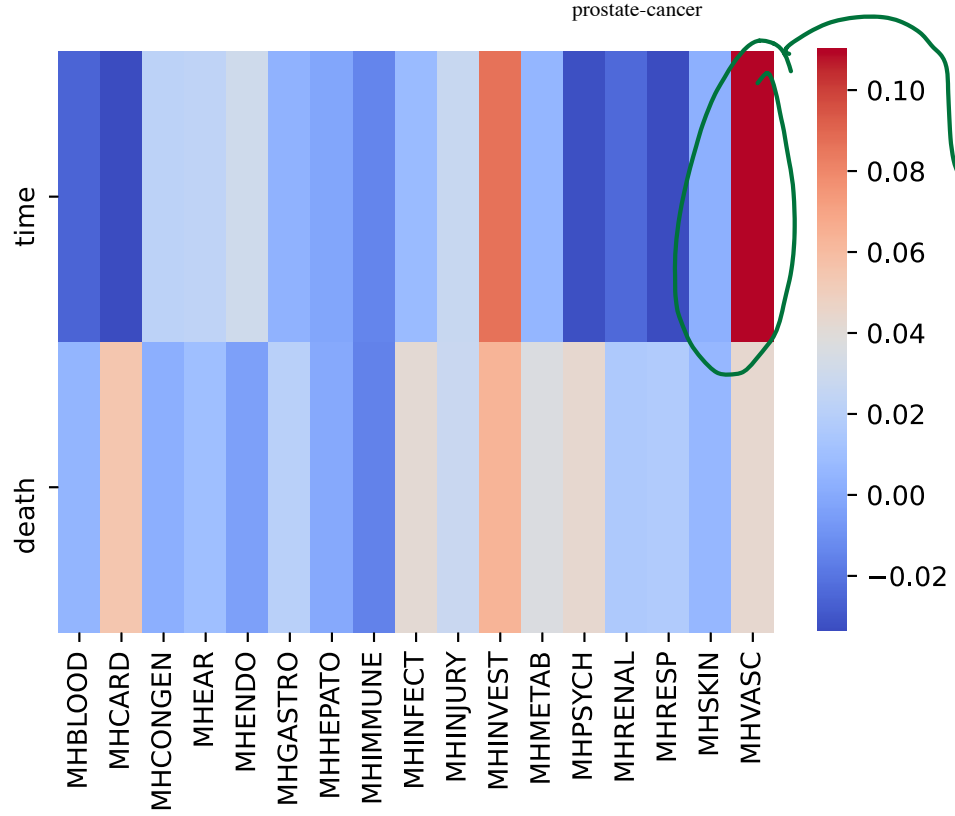
In [220…
```
data.corr()[mhBody].loc[['time','death']]
```

Out[220…

	MHBLOOD	MHCARD	MHCONGEN	MHEAR	MHENDO	MHGASTRO	MHHEPATO	MHIMM
time	-0.025828	-0.033762	0.021613	0.023002	0.030149	0.003517	-0.002399	-0.01
death	0.004564	0.054803	0.001486	0.009272	-0.004673	0.020256	-0.000465	-0.01

In [221…
```
sns.heatmap(data.corr()[mhBody].loc[['time','death']], cmap='coolwarm')
```

Out[221… <AxesSubplot:>



Baseline Patient Performance Status

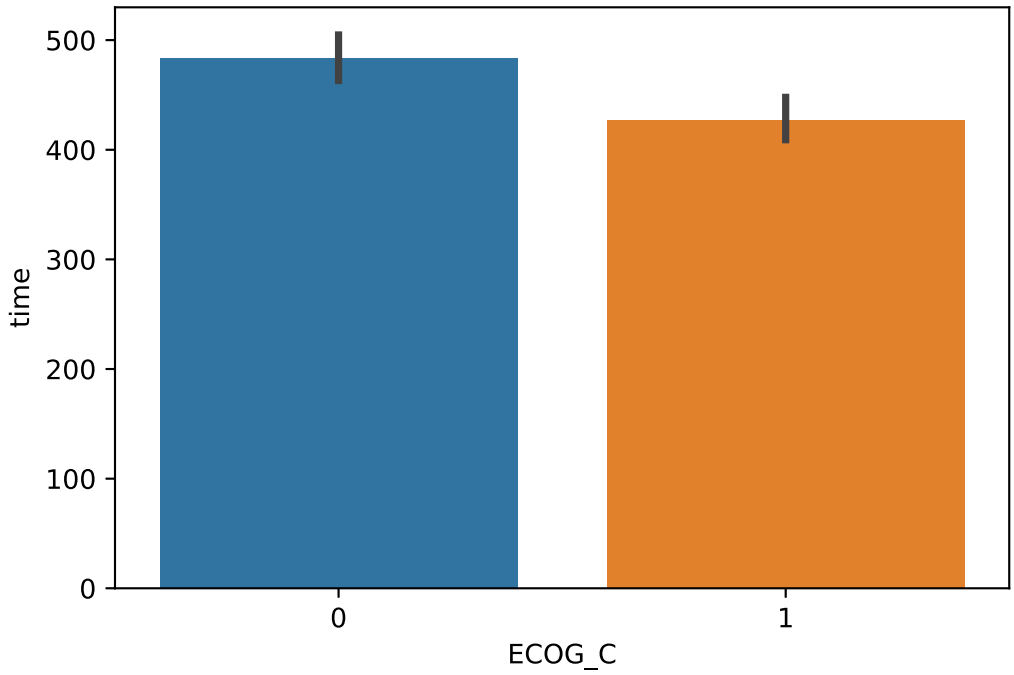
In [224…
```
data.corr()['ECOG_C'].loc[['time','death']]
```

Out[224…
```
time    -0.096396
death    0.109507
Name: ECOG_C, dtype: float64
```

In [227…
```
sns.barplot(x='ECOG_C',y='time',data=data)
```

Out[227…
```
<AxesSubplot:xlabel='ECOG_C', ylabel='time'>
```

balanced split



Race

In [229…
```
race = ['RaceAsian',
 'RaceBlack',
 'RaceOther',
 'RaceWhite']
```

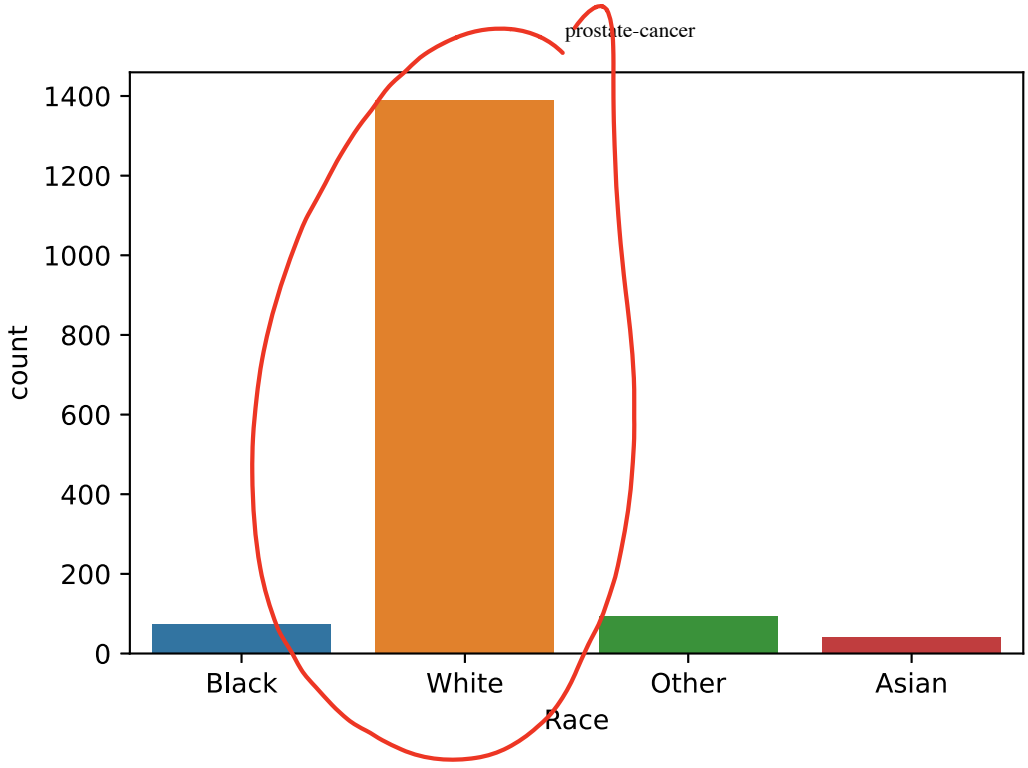
In [259…
```
def get_race(data):
    if data['RaceWhite'] == 1:
        return 'White'
    elif data['RaceBlack'] == 1:
        return 'Black'
    elif data['RaceAsian'] == 1:
        return 'Asian'
    else:
        return 'Other'
```

change dummy variable back to categorical

In [260…
```
data['Race'] = data[race].apply(get_race, axis=1)
```

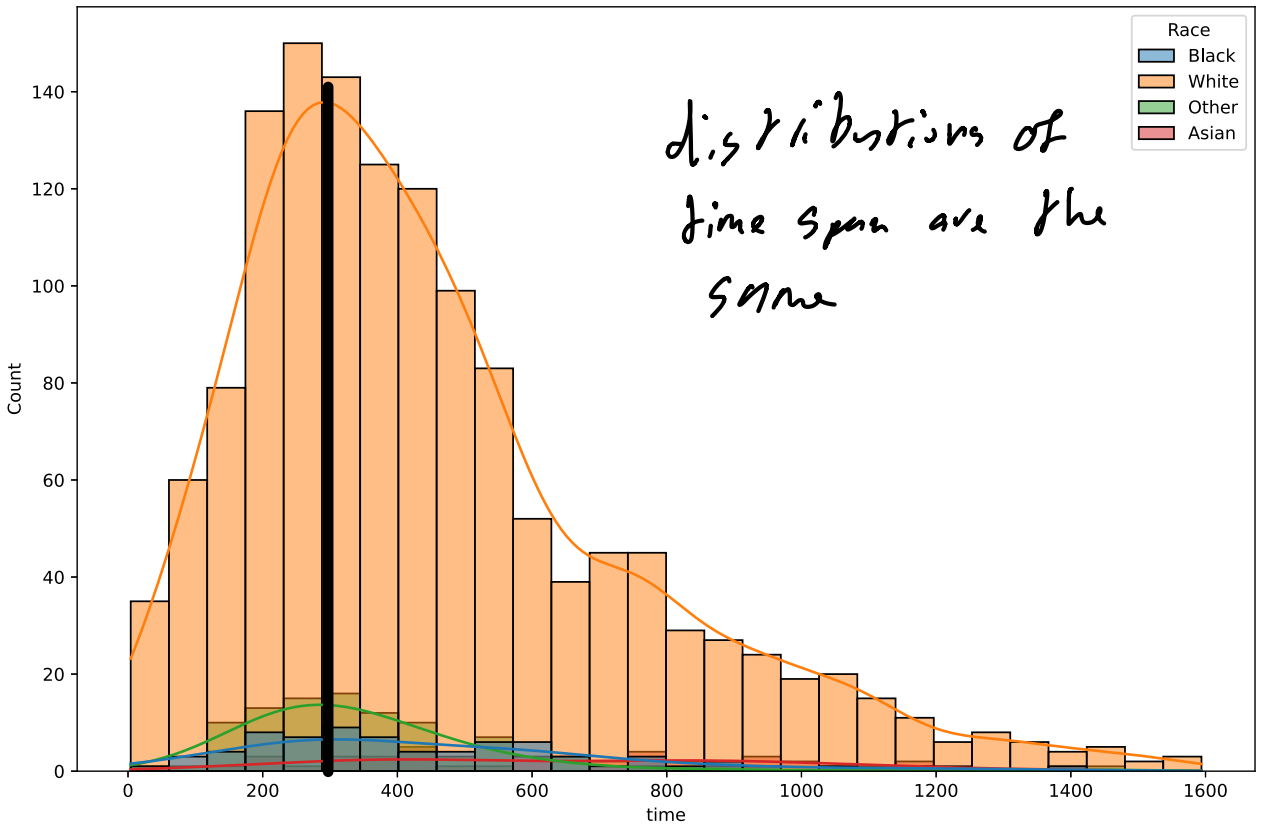
In [274…
```
sns.countplot(x='Race',data=data)
```

Out[274… <AxesSubplot:xlabel='Race', ylabel='count'>



In [277…
```
plt.figure(figsize=(12,8))
sns.histplot(x='time',data=data, hue='Race', kde=True)
```

Out[277… <AxesSubplot:xlabel='time', ylabel='Count'>



In [291…
```
data.corr()[race].loc[['time','death']]
```

Out[291…

	RaceAsian	RaceBlack	RaceOther	RaceWhite

	RaceAsian	RaceBlack	RaceOther	RaceWhite
time	0.112981	0.000800	-0.009367	0.008313
death	0.080378	-0.034214	-0.057938	0.060187

↳ significant correlation, but based on limited data

Region

In [278…
```
region = ['RegionAsia',
 'RegionEastEuro',
 'RegionNorthAmer',
 'RegionSouthAmer',
 'RegionWestEuro']
```

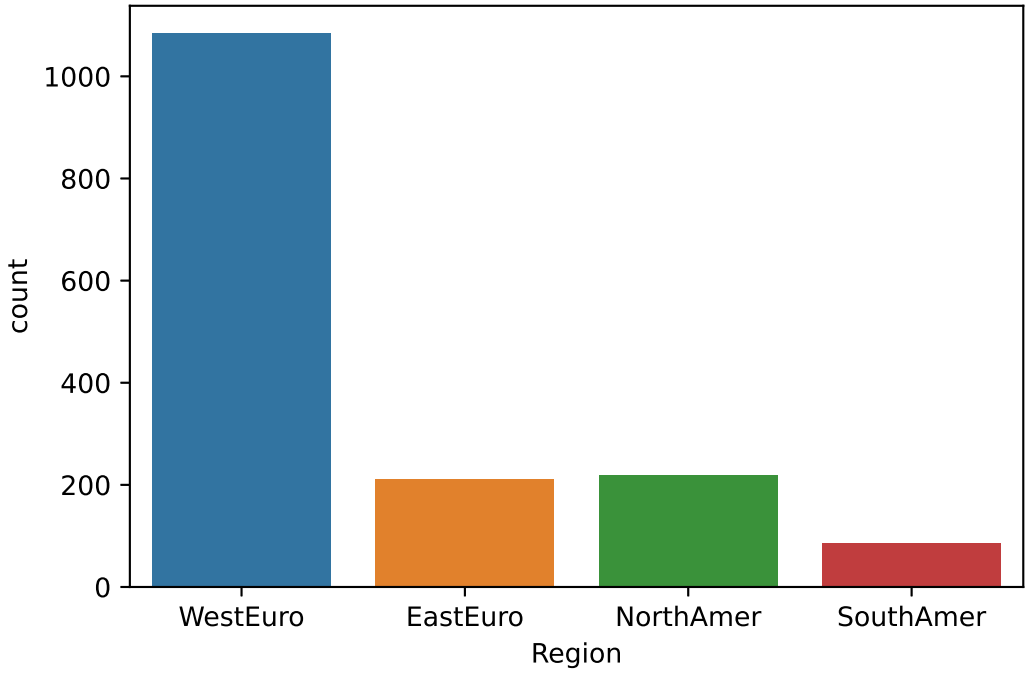
In [279…
```
def get_region(data):
    if data['RegionAsia'] == 1:
        return 'Asia'
    elif data['RegionEastEuro'] == 1:
        return 'EastEuro'
    elif data['RegionNorthAmer'] == 1:
        return 'NorthAmer'
    elif data['RegionSouthAmer'] == 1:
        return 'SouthAmer'
    else:
        return 'WestEuro'
```

change dummy variables back to categorical

In [280…
```
data['Region'] = data[region].apply(get_region, axis=1)
```

In [281…
```
sns.countplot(x='Region',data=data)
```

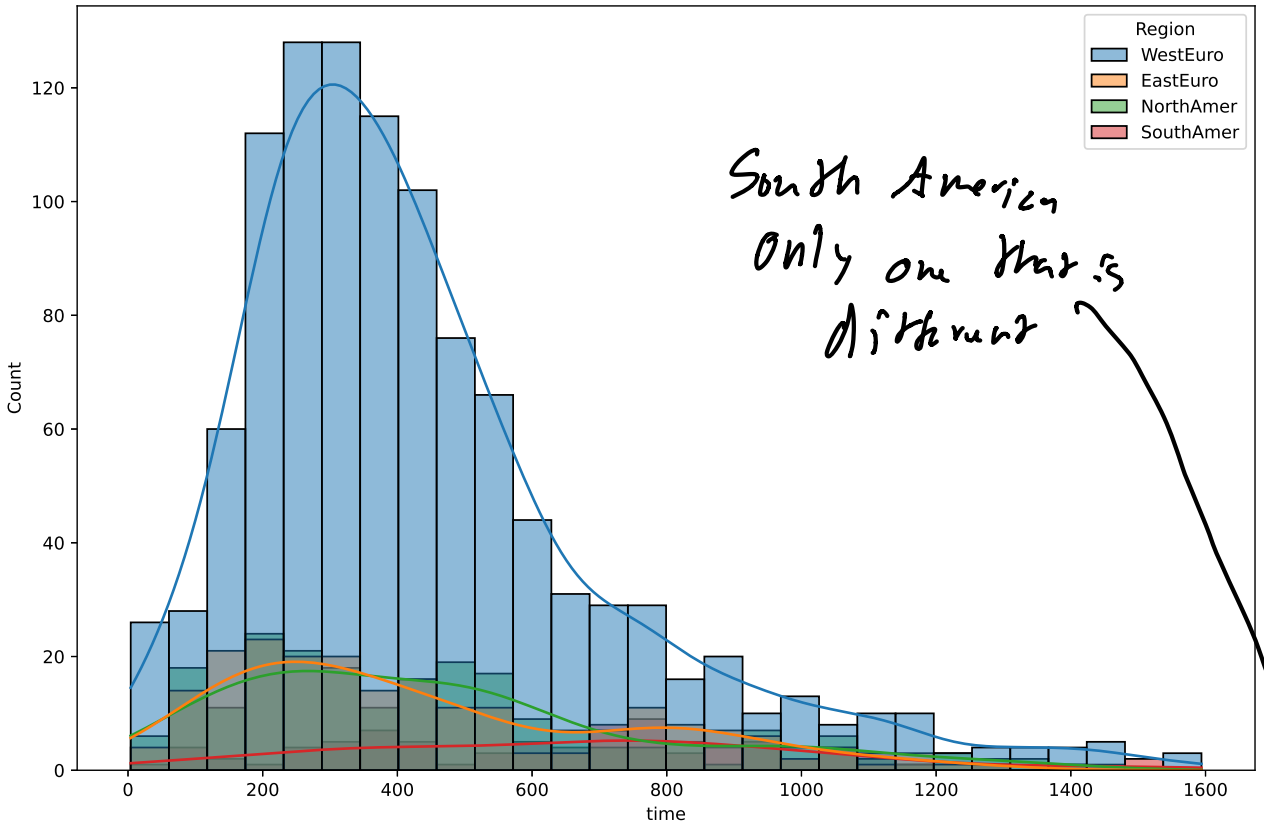
Out[281… <AxesSubplot:xlabel='Region', ylabel='count'>



- not even split
- very little info on South America
- nothing in Asia

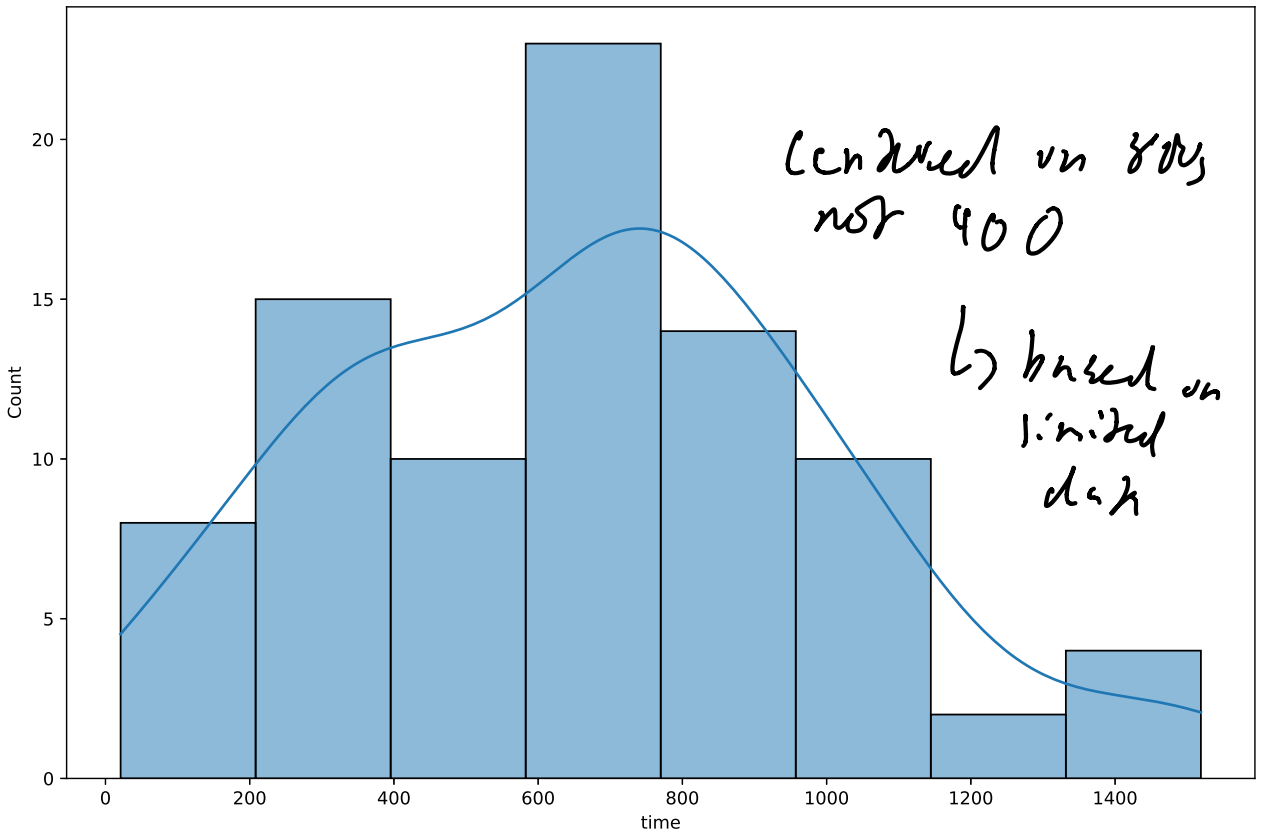
In [289…
```
plt.figure(figsize=(12,8))
sns.histplot(x='time',data=data, hue='Region', kde=True)
```

Out[289… <AxesSubplot:xlabel='time', ylabel='Count'>



In [307…
```
plt.figure(figsize=(12,8))
sns.histplot(x='time',data=data[data['Region'] == 'SouthAmer'], kde=True)
```

Out[307… <AxesSubplot:xlabel='time', ylabel='Count'>



In [294…
```
data.corr()[region].loc[['time','death']]
```

Out[294…

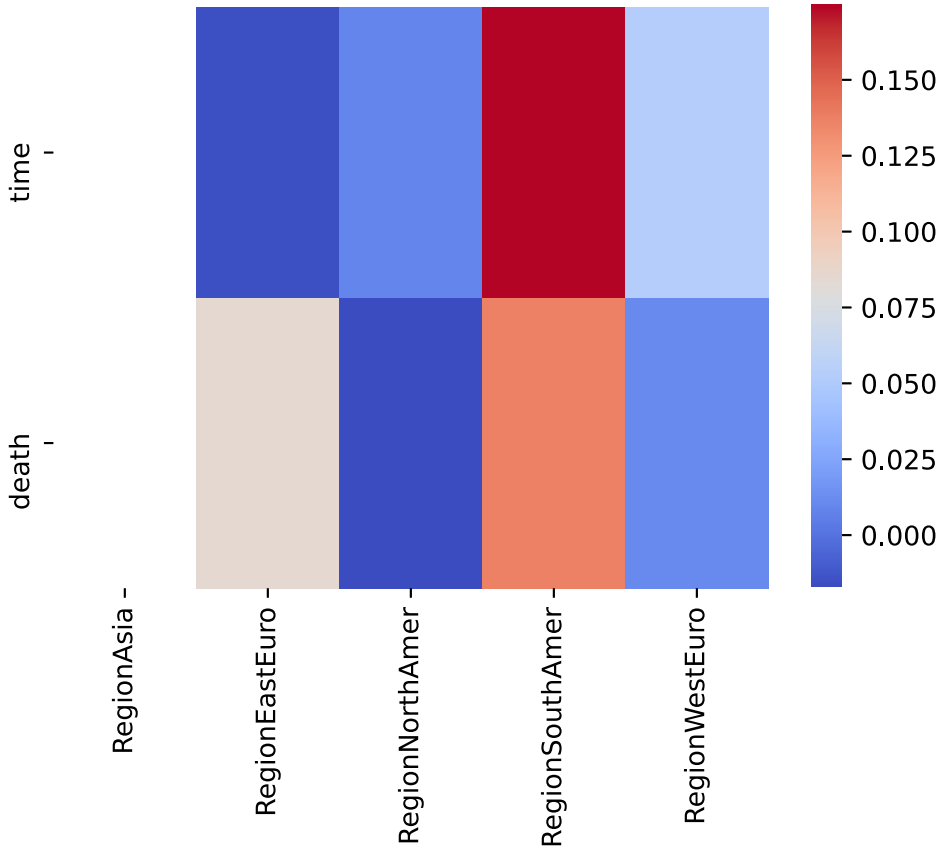
	RegionAsia	RegionEastEuro	RegionNorthAmer	RegionSouthAmer	RegionWestEuro
time	NaN	-0.015503	0.008615	0.173898	0.052483
death	NaN	0.084620	-0.017527	0.137065	0.010665

In [296…
```
data[region].sum()
```

Out[296…
```
RegionAsia          0
RegionEastEuro    211
RegionNorthAmer   219
RegionSouthAmer    86
RegionWestEuro    459
dtype: int64
```

In [301…
```
sns.heatmap(data.corr()[region].loc[['time','death']],cmap='coolwarm')
```

Out[301…
```
<AxesSubplot:>
```



In [306…
```
data.corr()['time'].sort_values(ascending=False)
```

Out[306…
```
time                1.000000
CORTICOSTEROID      0.205400
RegionSouthAmer     0.173898
TBILI               0.160275
HB                  0.149179
                      ...
NEUperLEU          -0.134277
LDH                -0.141076
ALP                -0.152988
TESTO              -0.207605
RegionAsia               NaN
Name: time, Length: 103, dtype: float64
```

In [ ]: