IAU Projekt 1.fáza

Autori: Peter Plevko (50%), Radovan Cyprich (50%)

Dátum: 30.10.2021

In [2]:

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib as mat
import matplotlib.pylab as pylab
import statsmodels.api as sm
import statsmodels.stats as sm_stats
import statsmodels.stats.api as sms
import scipy.stats as stats
from matplotlib import pyplot
from collections import Counter
```

Načítanie údajov z datasetov

In [3]:

```
filename = "061/profiles.csv"
profiles = pd.read_csv(filename, sep='\t')
profiles.head()
```

Out[3]:

	Unnamed: 0	residence	ssn	blood_group	birthdate	address	nam
0	0	55180 Charlotte Mission\nPort Janiceton, MD 01615	707-91-3436	AB+	12/25/1983, 00:00:00	2206 Regina Bypass Apt. 478\nSouth Samanthabur...	Juli Santiag
1	1	4157 Chelsea Extension Apt. 138\nPhillipstown,...	882-73-6960	B+	05/21/1936, 00:00:00	5495 James Causeway\nLake Charles, NC 64584	Brend Roac
2	2	8890 Rogers Trail\nNew April, VT 56293	395-27-1265	A+	05/11/1908, 00:00:00	USNV Olsen\nFPO AE 06714	Richar Thoma
3	3	6073 Roger Via Suite 739\nPort Johnfort, NM 79606	708-36-7168	B+	04/22/1907, 00:00:00	461 Jeffrey Tunnel\nPort Meganmouth, OK 29371	Angel Richard
4	4	6685 Jason Trafficway Apt. 492\nWest Deantown,...	183-78-8749	A-	1905-10-14	501 Johnson Roads Apt. 644\nJonathanborough, N...	Josep Elliot

In [4]:

```
filename2 = "061/labor.csv"
labor = pd.read_csv(filename2, sep='\t')
labor.head()
```

Out[4]:

	Unnamed: 0	relationship	smoker	alp	weight	trombocyty	hematokrit	hemoglobin	
0	0	divoced	yes	69.29754	96.36107	6.16009	5.43057	5.33106	5
1	1	separated	yes	77.14564	87.14201	7.56648	5.49149	6.93812	6
2	2	single	no	80.33335	77.58017	6.40181	3.43240	6.56676	5
3	3	divoced	no	87.09655	112.49541	6.50980	4.31559	6.39304	6
4	4	nop	yes	68.78724	85.92167	8.61102	7.98429	5.27338	8

Základný opis dát spolu s ich charakteristikami

DATASET PROFILES

Dataset *profiles.csv* obsahuje dokopy 3058 záznamov základných informácii o pacientoch. Počet všetkých atribútov týchto záznamov je 10. Atribút, ktorý nám hovorí o poradí záznamu nebudeme potrebovať, tým pádom tento stĺpec odstránime.

Jeden z týchto atributov je poradové číslo záznamu. Predpokladáme, že tento atribút reálne pri našom skúmaní a analýze využívať nebudeme, čiže bude vhodné ho z datasetu odstrániť ako aj ďalšie nepotrebné atribúty.

Zoznam jednotlivých atribútov:

- Unnamed: 0 - atribút celočíselná hodnota, reprezentujúca poradezáznamu.
- Residence - trvalé bydlisko
- SSN - sociálne bezpečnostné číslo
- Blood_group - krvná skupina subjektu
- Birthdate - dátum narodenia subjektu
- Address - adressa subjektu
- Name - meno a priezvisko subjektu
- Race - rasa subjektu
- Job - zamestnanie subjektu
- Sex - pohlavie (M - male alebo F - female)

In [5]:

```
profiles.describe()
```

Out[5]:

	Unnamed: 0
count	3058.000000
mean	1528.500000
std	882.912887
min	0.000000
25%	764.250000
50%	1528.500000
75%	2292.750000
max	3057.000000

In [6]:

```
len(profiles)
```

Out[6]:

```
3058
```

In [7]:

```
len(profiles.columns)
```

Out[7]:

```
10
```

In [8]:

```
profiles.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3058 entries, 0 to 3057
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Unnamed: 0   3058 non-null   int64 
 1   residence    3058 non-null   object
 2   ssn          3058 non-null   object
 3   blood_group  3058 non-null   object
 4   birthdate    3058 non-null   object
 5   address      3058 non-null   object
 6   name         3058 non-null   object
 7   race         3058 non-null   object
 8   job          3058 non-null   object
 9   sex          3058 non-null   object
dtypes: int64(1), object(9)
memory usage: 239.0+ KB
```

Identifikácia problémov v dátach s navrhnutým riešením

V datasete profiles sa nachádza niekoľko nezrovnalostí:

- Nepotrebné atribúty
- Overenie duplicitných záznamov
- Chybné záznamy v jednotlivých stĺpcoch
- Nejednotná sytnax pre rovnaké výpovedné hodnoty atribútov

Preto si najskôr tieto chyby odstránime.

In [9]:

```
profiles = profiles.iloc[: , 1:]
```

Ako prvé si z datasetu odstránime nepotrebný atribút *Unnamed*, ktorý nám len označuje index záznamu. Vymažeme si ho aj kvôli tomu aby sa nám podarilo zistiť, či sa v datasete nachádzajú nejaké duplicitné záznamy.

In [10]:

```
duplicate = profiles[profiles.duplicated()]
profiles = profiles.drop_duplicates()
print("Počet záznamov v datasete: "+str(len(profiles)))
print("V datasete sa nachádza: " + str(len(duplicate)) + " duplikátov.")
```

```
Počet záznamov v datasete: 3058
V datasete sa nachádza: 0 duplikátov.
```

Ako môžme vidieť dĺžka datasetu sa nezmenila, čo znamená, že nemáme úplne identické riadky. Avšak môže sa stať, že niektorí ľudia budú mať rovnaké meno ale ostatné atribúty sa budú líšiť.

In [11]:

```
def monthToNum(shortMonth):
    return {
            'Jan': '01',
            'Feb': '02',
            'Mar': '03',
            'Apr': '04',
            'May': '05',
            'Jun': '06',
            'Jul': '07',
            'Aug': '08',
            'Sep': '09',
            'Oct': '10',
            'Nov': '11',
            'Dec': '12'
    }[shortMonth]

for x in profiles['birthdate']:

    # nahradim nulovy datum
    if(len(x)==20):
        removedZeros = x.replace(", 00:00:00", "")
        array = removedZeros.split("/")
        newString = array[2] + "-" + array[0] + "-" + array[1]
        profiles['birthdate'] = profiles['birthdate'].replace(x, newString)

    # nahradim lomitkovy datum
    elif(len(x)==10):
        profiles['birthdate'] = profiles['birthdate'].replace(x, x.replace("/", "-"))

    # menim slovo mesiac na cislo
    elif(len(x)==11):
        array = x.split(" ")
        newString = array[2] + "-" + monthToNum(array[1]) + "-" + array[0]
        profiles['birthdate'] = profiles['birthdate'].replace(x, newString)
```

Dátumy v datasete sú zaznamenávané v rôznych formátoch, preto sme si určili jednotný formát dátumu a upravili sme všetky záznamy podľa tohto formátu.

In [12]:

```
print(profiles.race.unique())
```

```
['White' 'Black' 'white' 'Asian' 'black' 'Hawaiian' 'Indian' 'blsck']
```

In [13]:

```
profiles['race'] = profiles['race'].astype(str).str.replace('blsck',"Black")
profiles['race'] = profiles['race'].astype(str).str.replace('black',"Black")
profiles['race'] = profiles['race'].astype(str).str.replace('white',"White")
```

V stĺpci *race* sa nachádzalo pár chýb preto sme si tieto chyby taktiež na začiatku upravili na jednotné hodnoty.

Zaujímavé atribúty

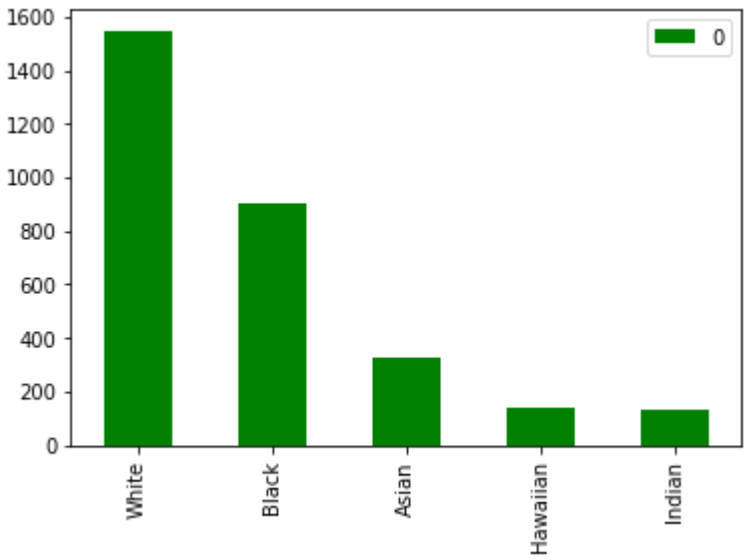
Atribúty z datasetu *profiles.csv*, ktoré nás budú zaujímať sú najmä pohlavie, rasa a krvná skupina jedincov. Pri skúmaní môžeme napríklad zisťovať, či existuje nejaká závislosť medzi krvnou skupinou, pohlavím a rasou ľudí s nábehom na leukémiu. Atribúty name a address budeme potrebovať pre vyhľadávanie hodnôt v druhom datasete *labor.csv*, keďže tieto atribúty sa nachádzajú v oboch z týchto datasetov.

In [14]:

```
count = Counter(profiles.race)
df = profiles.from_dict(count, orient='index')
df.plot(kind='bar',color='g')
```

Out[14]:

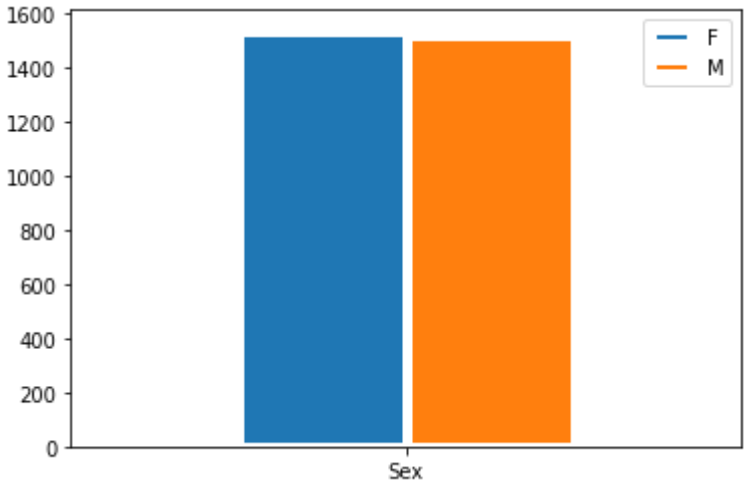
```
<AxesSubplot:>
```



In [15]:

```
count = Counter(profiles.sex)
df = profiles.from_dict(count, orient='index')
df = pd.DataFrame(count, index=['Sex'])
ax = df.plot.bar(rot=0,edgecolor='white', linewidth=5)
print(count)
```

```
Counter({'F': 1537, 'M': 1521})
```



In [16]:

```
x=profiles.sex[(profiles["sex"] == "M") | (profiles["sex"] == "F")].count()
print(x)
```

```
3058
```

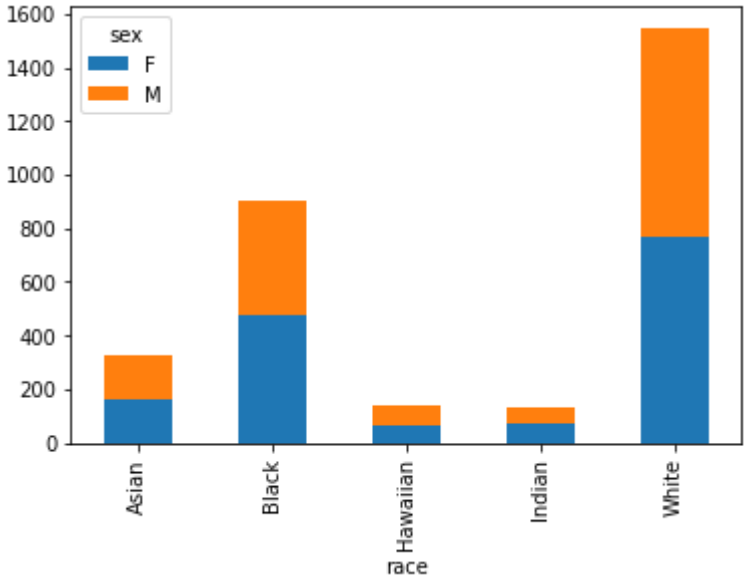
Počet stĺpcov, v ktorých sa nachádza informácia o pohlaví sa rovná počtu všetkých záznamov. Tým pádom sa tu nenachádzajú žiadne chýbajúce hodnoty.

In [17]:

```
pd.crosstab(index=profiles["race"], columns=profiles["sex"]).plot.bar(stacked=True)
```

Out[17]:

```
<AxesSubplot:xlabel='race'>
```

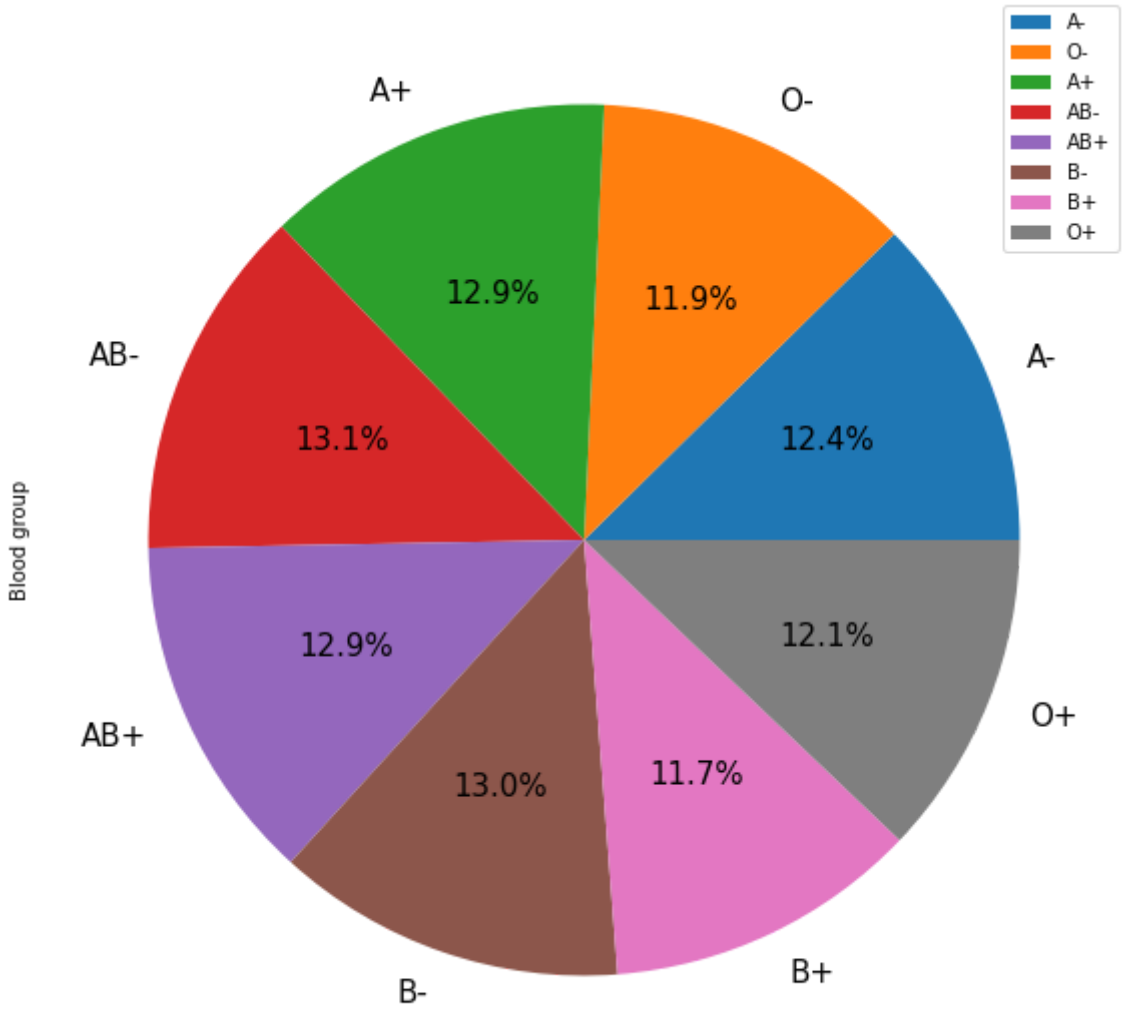


Tento graf zobrazuje koľko osôb z danej rasy ľudí je mužov a koľko žien. Teda môžme vidieť, že počty sa príliš nelíšia a sú približne rovnaké.

In [18]:

```
count = Counter(profiles.blood_group)
values=count.values()
index=sorted(count, key=count.get, reverse=True)
df = pd.DataFrame({'Blood group': values}, index=index)
df.plot(kind='pie',y='Blood group',figsize=(15,10), fontsize = 15,autopct='%1.1f%%')
print(count)
```

```
Counter({'A-': 400, 'O-': 397, 'A+': 396, 'AB-': 396, 'AB+': 379, 'B-': 370,
'B+': 363, 'O+': 357})
```



DATASET LABOR

Dataset *labor.csv* obsahuje dokopy 10041 záznamov o pacientoch. Počet všetkých atribútov týchto záznamov je 18. Atribút, ktorý nám hovorí o poradí záznamu nebudeme potrebovať, tým pádom tento stĺpec odstránime. Unnamed: 0 - atribút, hovoriaci o poradí záznamu.

Zoznam jednotlivých atribútov:

- Unnamed: 0 - atribút celočíselná hodnota, reprezentujúca poradezáznamu.
- Relationship - atribút o partnerskom stave subjektu
- Smoker - atribút hovoriaci o tom, či subejkt fajčí
- Alp - atribút reprezentujúci hodnotu enzýmu alkaline phosphatase v krvi subjektu
- Weight - váha subjektu
- Trombocyty - množstvo krvných doštičiek v krvi subjektu
- Hematokrit - podiel červených krviniek na celkový objem krvi subjektu
- Hemoglobin - hodnota červeného krvného farbiva
- er-cv - Cardiovascular risk profile in patients with estrogen receptor
- Ast - atribút reprezentujúci hodnotu enzýmu aspartate aminotransferase v krvi subjektu
- SSN - sociálne bezpečnostné číslo subjektu
- Erytrocyty - hodnota červených krviniek v krvi subjektu
- Indicator - určuje stav pacienta a teda potreba ďalšieho vyšetrenia subjektu
- Hbver - hodnota nosiča vírusu hepatitídy typu B
- Leukocyty - hodnota bielych krviniek v krvi subjektu
- Alt - atribút reprezentujúci hodnotu enzýmu alanin transaminase v krvi subjektu
- Etytr - neznámy atribút
- Name - meno subjektu

In [19]:

```
labor.describe()
```

Out[19]:

	Unnamed: 0	alp	weight	trombocyty	hematokrit	hemoglobin	
count	10041.00000	10011.000000	10041.000000	10011.000000	10011.000000	10009.000000	100
mean	5020.00000	55.406927	70.201047	7.312738	5.691612	6.145086	5
std	2898.73136	25.893033	34.508025	1.059516	1.553295	1.671431	1
min	0.00000	0.000000	-54.300060	2.982940	0.421130	1.096590	
25%	2510.00000	32.751965	46.899880	6.626960	4.523520	4.883300	4
50%	5020.00000	60.544470	69.987990	7.316000	5.642440	6.108760	5
75%	7530.00000	78.923650	93.182390	8.008705	6.923710	7.392180	6
max	10040.00000	100.000000	228.356350	11.916140	9.927910	12.703130	10

In [20]:

```
len(labor)
```

Out[20]:

```
10041
```

In [21]:

```
len(labor.columns)
```

Out[21]:

```
18
```

In [22]:

```
labor.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10041 entries, 0 to 10040
Data columns (total 18 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0    10041 non-null  int64  
 1   relationship  10041 non-null  object 
 2   smoker        10041 non-null  object 
 3   alp           10011 non-null  float64
 4   weight        10041 non-null  float64
 5   trombocyty    10011 non-null  float64
 6   hematokrit    10011 non-null  float64
 7   hemoglobin    10009 non-null  float64
 8   er-cv         10011 non-null  float64
 9   ast           10011 non-null  float64
 10  ssn           10041 non-null  object 
 11  erytrocyty    10010 non-null  float64
 12  indicator     10041 non-null  float64
 13  hbver         10011 non-null  float64
 14  leukocyty     10010 non-null  float64
 15  alt           10011 non-null  float64
 16  etytr         10011 non-null  float64
 17  name          10041 non-null  object 
dtypes: float64(13), int64(1), object(4)
memory usage: 1.4+ MB
```

Identifikácia problémov v dátach s navrhnutým riešením

V datasete labor sa taktiež nachádza pár chýb, ktoré je vhodné eliminovať pred analýzou:

- Nepotrebné atribúty
- Overenie duplicitných záznamov
- Chybné záznamy v jednotlivých stĺpcoch
- Chýbajúce hodnoty
- Nejednotná sytnax pre rovnaké výpovedné hodnoty atribútov

In [23]:

```
labor = labor.iloc[: , 1:]
```

Ako prvé si z datasetu odstránime nepotrebný atribút *Unnamed*, ktorý nám len označuje index záznamu. Vymažeme si ho aj kvôli tomu aby sa nám podarilo zistiť, či sa v datasete nachádzajú nejaké duplicitné záznamy.

In [24]:

```
duplicates = labor[labor.duplicated()]
labor = labor.drop_duplicates()
print("Počet záznamov v datasete: "+str(len(labor)))
print("V datasete sa nachádza: " + str(len(duplicates)) + " duplikátov.")
```

```
Počet záznamov v datasete: 9942
V datasete sa nachádza: 99 duplikátov.
```

In [25]:

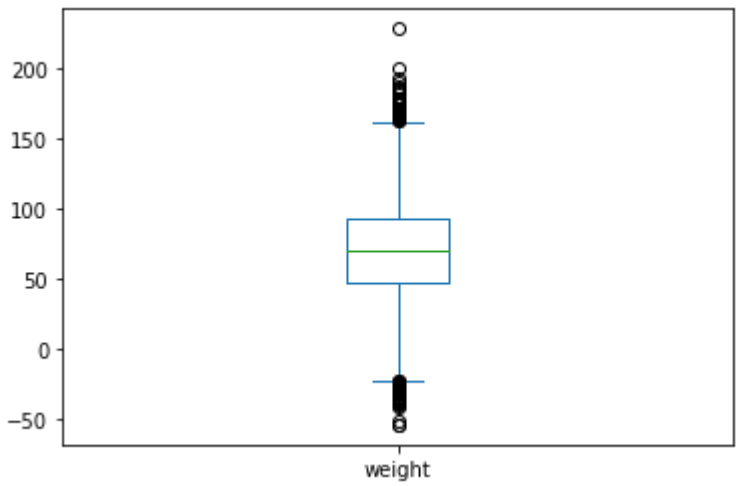
```
labor['relationship'] = labor['relationship'].str.replace('divoced',"divorced")
labor['smoker'] = labor['smoker'].str.replace('Y',"yes")
labor['smoker'] = labor['smoker'].str.replace('N',"no")
```

In [26]:

```
labor.weight.plot(kind='box')
```

Out[26]:

```
<AxesSubplot:>
```



V stĺpcoch *relationship* a *smoker* sú preklepy a nezhodný formát hodnôt, preto si ich nahradíme jednotnými hodnotami.

In [27]:

```
minusWeight = labor.loc[labor['weight'] < 3]
print(len(minusWeight))
index_names = labor[ labor['weight'] < 3 ].index
labor.drop(index_names, inplace = True)
```

```
240
```

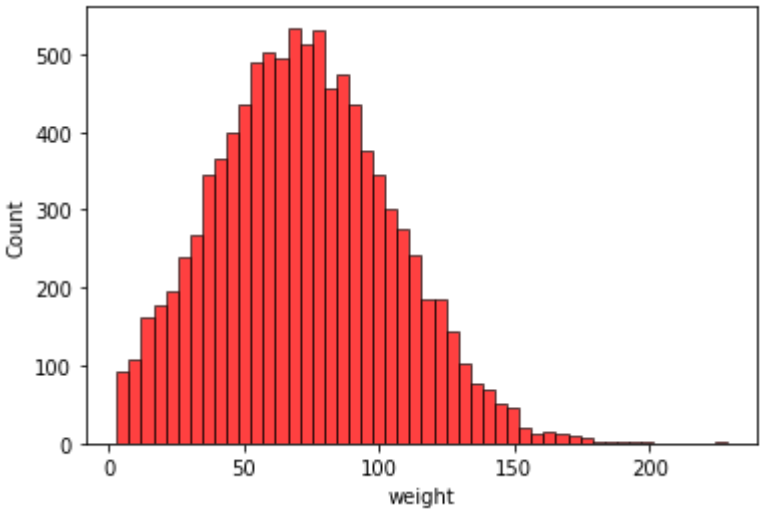
V stĺpci *weight*, ktorý reprezentuje hodnotu váhy človeka sme sa rozhodli vylúčiť hodnoty menšie ako 3, keďže novorodenci majú pri narodení okolo troch kíl. Počet týchto záznamov je 241.

In [28]:

```
sns.histplot(labor['weight'].dropna(),bins=50, color = "red")
```

Out[28]:

```
<AxesSubplot:xlabel='weight', ylabel='Count'>
```



In [29]:

```
labor.isnull().sum()
```

Out[29]:

```
relationship     0
smoker           0
alp             30
weight           0
trombocyty      28
hematokrit      28
hemoglobin      30
er-cv           30
ast             29
ssn              0
erytrocyty      29
indicator        0
hbver           29
leukocyty       30
alt             29
etytr           30
name             0
dtype: int64
```

Ako máme možnosť vidieť v niektorých stĺpcoch chýbajú viaceré hodnoty. Preto sme sa rozhodli tieto číselné atribúty nahradiť priemerom, keďže nechceme prísť o ďalšie dáta.

In [30]:

```
altMean = labor['alt'].mean()
labor['alt'] = labor['alt'].fillna(altMean)
etytrMean = labor['etytr'].mean()
labor['etytr'] = labor['etytr'].fillna(etytrMean)
hbverMean = labor['hbver'].mean()
labor['hbver'] = labor['hbver'].fillna(hbverMean)
leukocytyMean = labor['leukocyty'].mean()
labor['leukocyty'] = labor['leukocyty'].fillna(leukocytyMean)
erytrocytyMean = labor['erytrocyty'].mean()
labor['erytrocyty'] = labor['erytrocyty'].fillna(erytrocytyMean)
ercvMean = labor['er-cv'].mean()
labor['er-cv'] = labor['er-cv'].fillna(ercvMean)
astMean = labor['ast'].mean()
labor['ast'] = labor['ast'].fillna(astMean)
hemoglobinMean = labor['hemoglobin'].mean()
labor['hemoglobin'] = labor['hemoglobin'].fillna(hemoglobinMean)
trombocytyMean = labor['trombocyty'].mean()
labor['trombocyty'] = labor['trombocyty'].fillna(trombocytyMean)
hematokritMean = labor['hematokrit'].mean()
labor['hematokrit'] = labor['hematokrit'].fillna(hematokritMean)
alpMean = labor['alp'].mean()
labor['alp'] = labor['alp'].fillna(alpMean)
```

In [31]:

```
labor.isnull().sum()
```

Out[31]:

```
relationship    0
smoker          0
alp             0
weight          0
trombocyty      0
hematokrit      0
hemoglobin      0
er-cv           0
ast             0
ssn             0
erytrocyty      0
indicator       0
hbver           0
leukocyty       0
alt             0
etytr           0
name            0
dtype: int64
```

Zaujímave atribúty

Atribúty z datasetu labor.csv, sú najmä číselné atribúty a preto viacero z nich je zaujímavých. Je možné pozorovať rôzne závislosti a vzťahy medzi nimi. Nás budú zaujímať hlavne atribúty:

- indicator
- trombocyty
- hemoglobin.
- erytrocyty
- alt
- hematokrit

Atribút name budeme potrebovať pre spojenie s druhým datasetom profiles.csv, keďže tento atribút je spoločný.

In [32]:

```
labor.indicator.describe()
```

Out[32]:

```
count    9702.000000
mean        0.646568
std         0.478060
min         0.000000
25%         0.000000
50%         1.000000
75%         1.000000
max         1.000000
Name: indicator, dtype: float64
```

In [33]:

```
labor['indicator'].value_counts()
```

Out[33]:

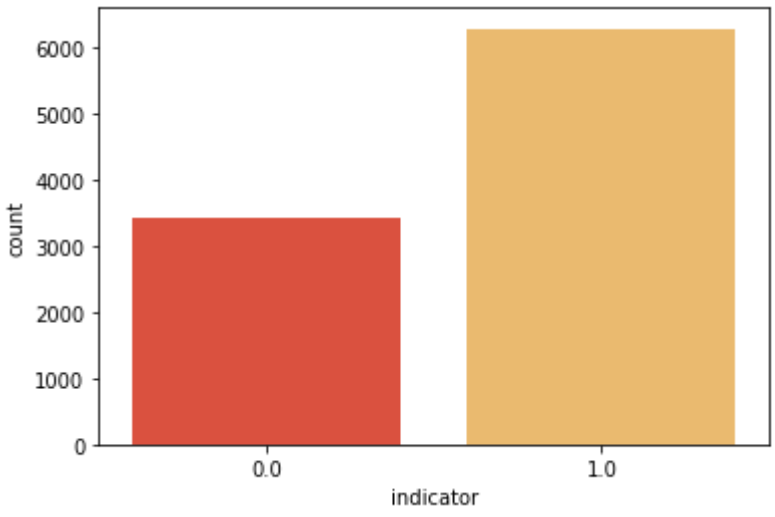
```
1.0    6273
0.0    3429
Name: indicator, dtype: int64
```

In [34]:

```
sns.countplot(data=labor,x="indicator",palette=("YlOrRd_r"))
```

Out[34]:

```
<AxesSubplot:xlabel='indicator', ylabel='count'>
```



In [35]:

```
labor.trombocyty.describe()
```

Out[35]:

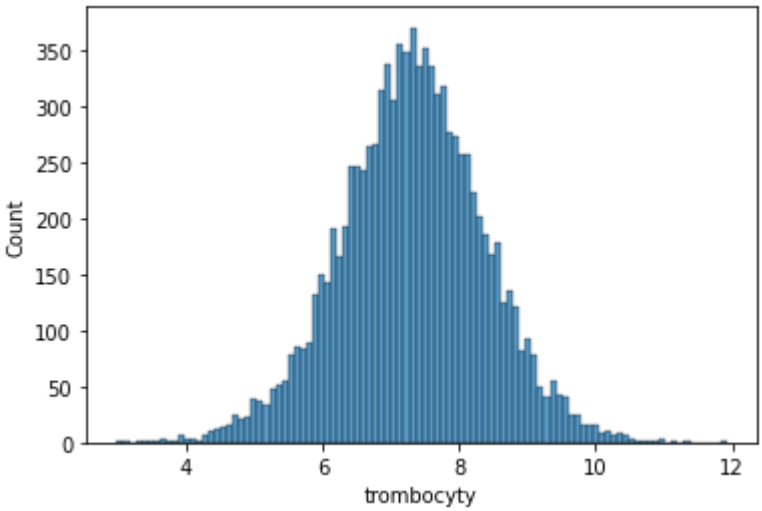
```
count    9702.000000
mean        7.313176
std         1.055974
min         2.982940
25%         6.630693
50%         7.314570
75%         8.003755
max        11.916140
Name: trombocyty, dtype: float64
```

In [36]:

```
sns.histplot(labor.trombocyty,bins=100)
```

Out[36]:

```
<AxesSubplot:xlabel='trombocyty', ylabel='Count'>
```

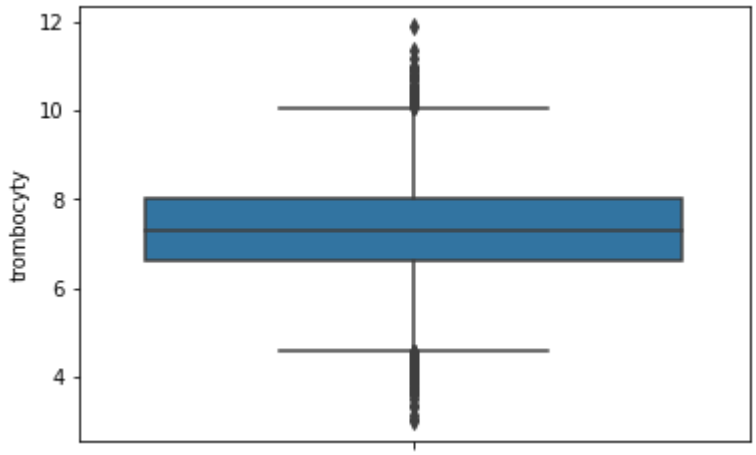


In [37]:

```
sns.boxplot(y = labor.trombocyty)
```

Out[37]:

```
<AxesSubplot:ylabel='trombocyty'>
```



In [38]:

```
labor.hemoglobin.describe()
```

Out[38]:

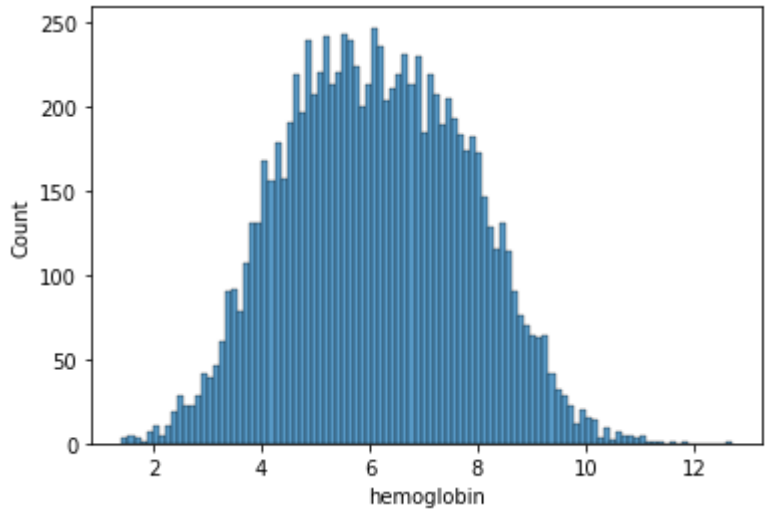
```
count    9702.000000
mean        6.143373
std         1.665373
min         1.416910
25%         4.887585
50%         6.114375
75%         7.384968
max        12.703130
Name: hemoglobin, dtype: float64
```

In [39]:

```
sns.histplot(labor.hemoglobin,bins=100)
```

Out[39]:

```
<AxesSubplot:xlabel='hemoglobin', ylabel='Count'>
```

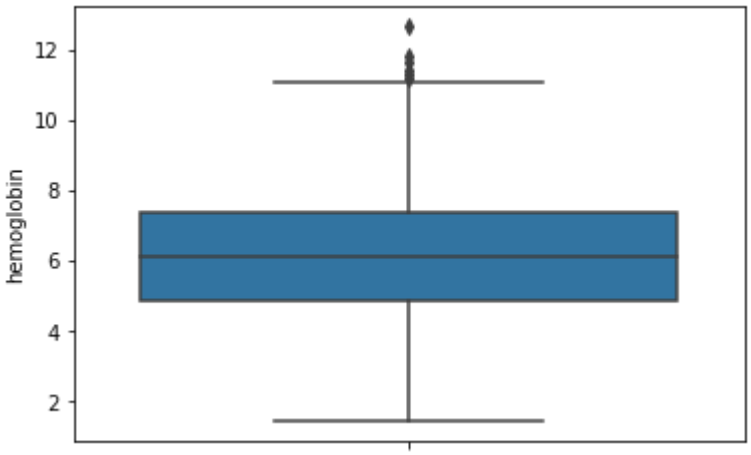


In [40]:

```
sns.boxplot(y = labor.hemoglobin)
```

Out[40]:

```
<AxesSubplot:ylabel='hemoglobin'>
```



In [41]:

```
labor.erytrocyty.describe()
```

Out[41]:

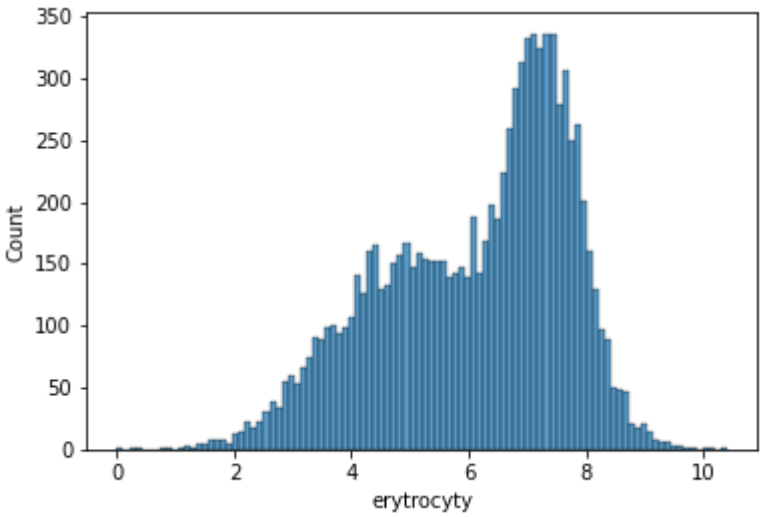
```
count    9702.000000
mean        6.114550
std         1.589420
min         0.000000
25%         4.909688
50%         6.518850
75%         7.366485
max        10.401110
Name: erytrocyty, dtype: float64
```

In [42]:

```
sns.histplot(labor.erytrocyty,bins=100)
```

Out[42]:

```
<AxesSubplot:xlabel='erytrocyty', ylabel='Count'>
```

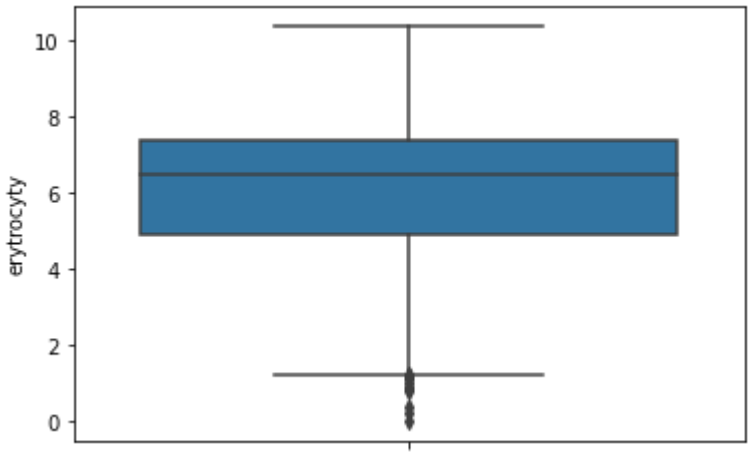


In [43]:

```
sns.boxplot(y = labor.erytrocyty)
```

Out[43]:

```
<AxesSubplot:ylabel='erytrocyty'>
```



Type *Markdown* and LaTeX: α^2

In [44]:

```
labor.alt.describe()
```

Out[44]:

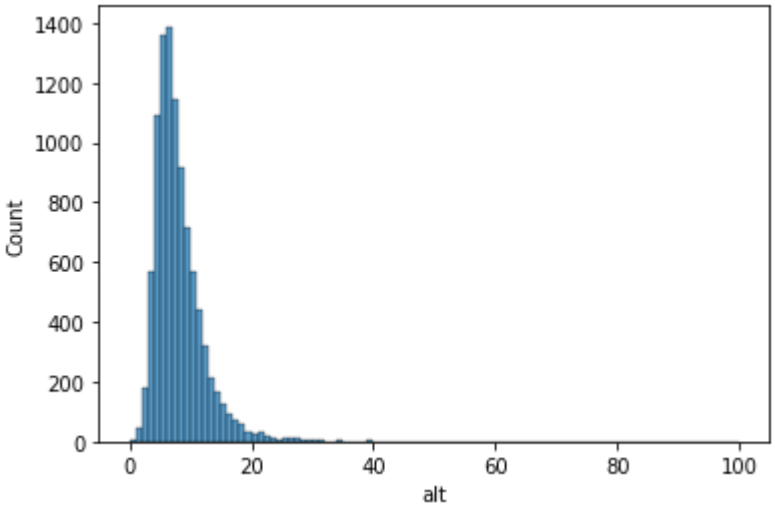
```
count    9702.000000
mean        8.165743
std         4.563609
min         0.000000
25%         5.413398
50%         7.173520
75%         9.772753
max       100.000000
Name: alt, dtype: float64
```

In [45]:

```
sns.histplot(labor.alt,bins=100)
```

Out[45]:

```
<AxesSubplot:xlabel='alt', ylabel='Count'>
```

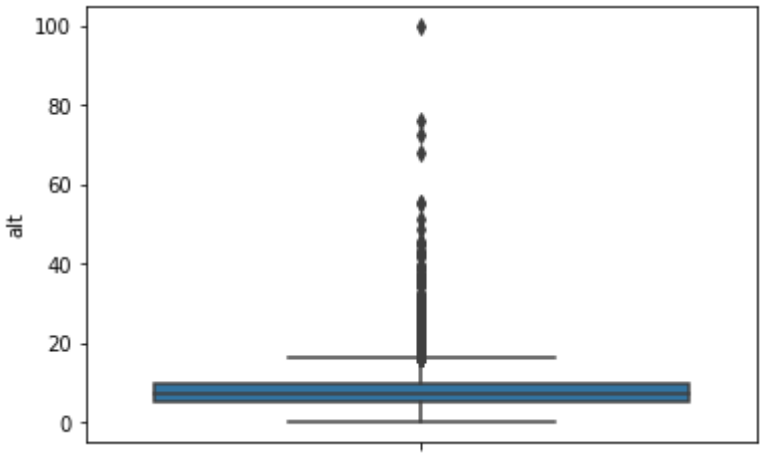


In [46]:

```
sns.boxplot(y=labor.alt)
```

Out[46]:

```
<AxesSubplot:ylabel='alt'>
```



In [47]:

```
labor.hematokrit.describe()
```

Out[47]:

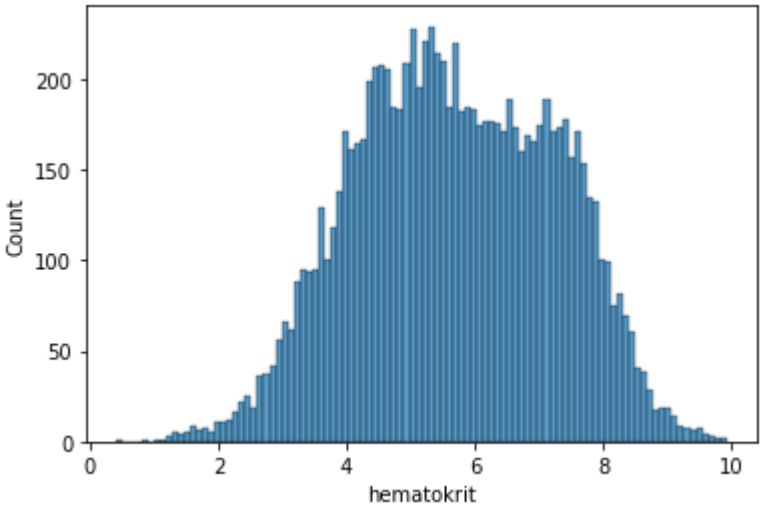
```
count    9702.000000
mean        5.691604
std         1.552859
min         0.421130
25%         4.529228
50%         5.646530
75%         6.923930
max         9.927910
Name: hematokrit, dtype: float64
```

In [48]:

```
sns.histplot(labor.hematokrit,bins=100)
```

Out[48]:

```
<AxesSubplot:xlabel='hematokrit', ylabel='Count'>
```

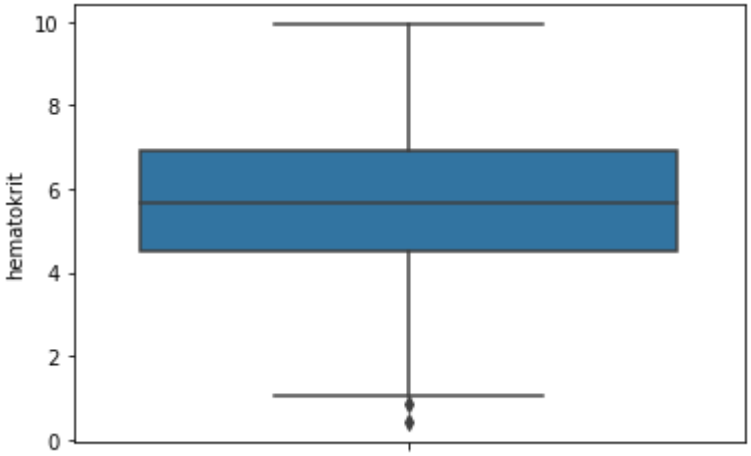


In [49]:

```
sns.boxplot(y=labor.hematokrit)
```

Out[49]:

```
<AxesSubplot:ylabel='hematokrit'>
```



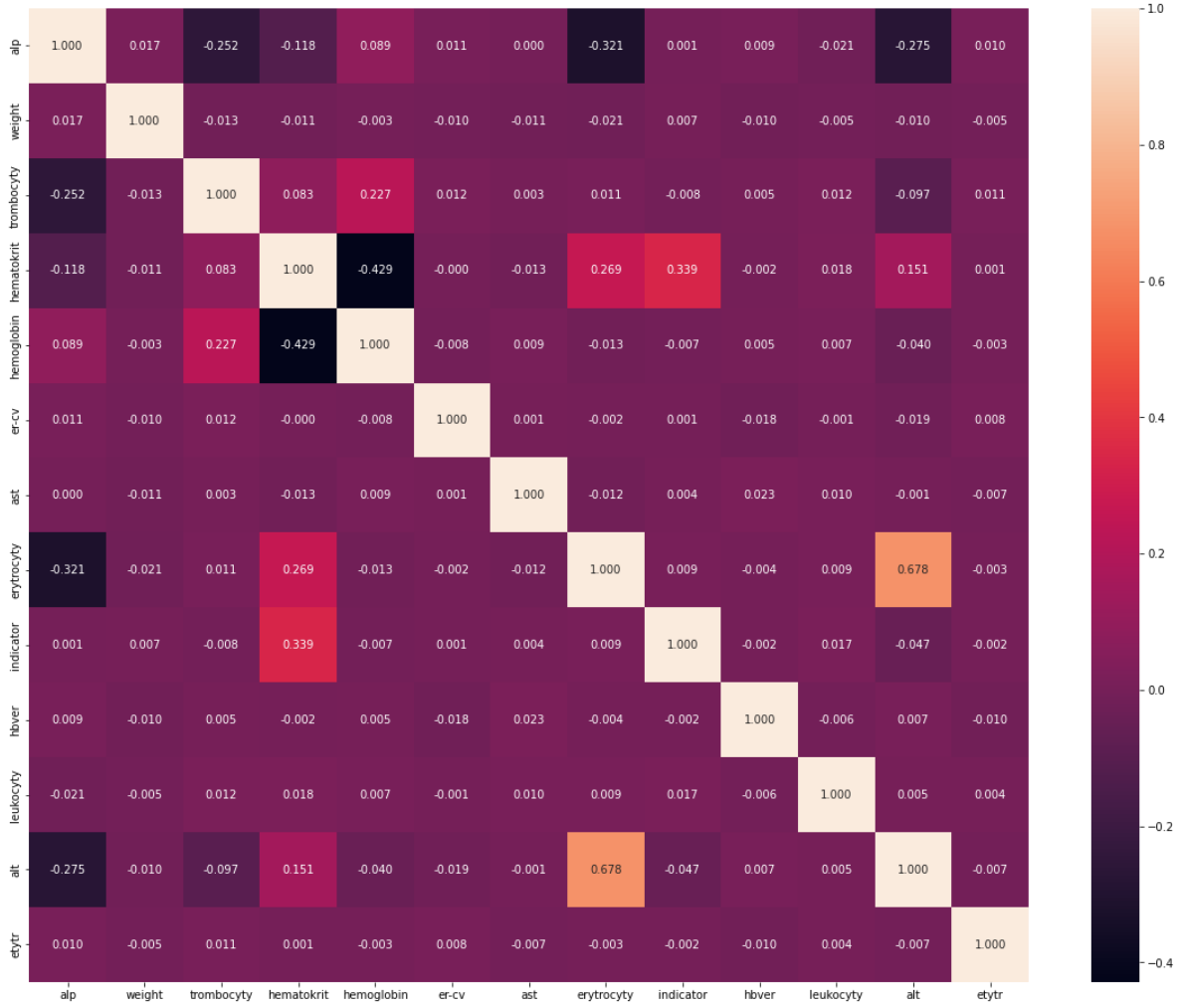
Párová analýza dát

In [50]:

```
figure, ax = plt.subplots(figsize=(20,16))
sns.heatmap(labor.corr(),ax=ax, annot =True, fmt = ".3f")
```

Out[50]:

<AxesSubplot:>



Z tejto heatmapy môžeme vyčítať viacero informácii. Ako máme možnosť vidieť atribúty:

- etytr
- leukocity
- hbver
- ast
- er-cv
- weight

Majú takmer nulovú koreláciu s ostatnými atribútmi. To znamená, že tieto atribúty nám vo vzájomných závislostiach neposkytujú takmer žiadnu informáciu. Naopak si môžeme všimnúť atribúty:

- erytrocyty
- hematokrit
- alt
- indicator

- trombocyty
- hemoglobin

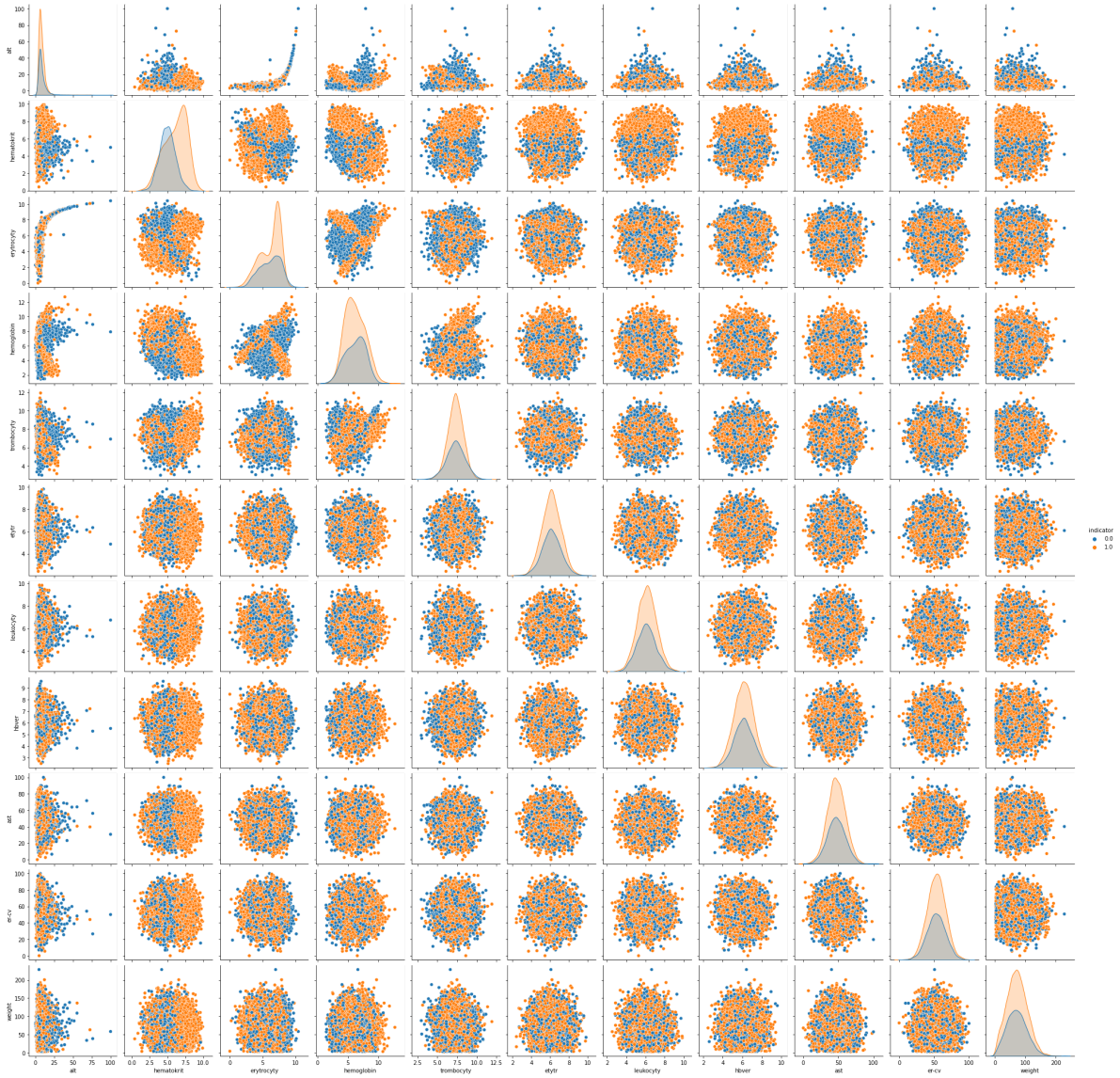
Medzi sebou v určitej miere korelujú a preto budeme skúmať ich závislosti neskôr.

In [51]:

```
sns.pairplot(labor, vars=['alt','hematokrit', 'erytrocyty','hemoglobin','trombocyty','etytr
```

Out[51]:

```
<seaborn.axisgrid.PairGrid at 0x2af7477dc40>
```



Postupne sme si vygenerovali všetky závislé grafy vzhľadom na atribút *indicator*.

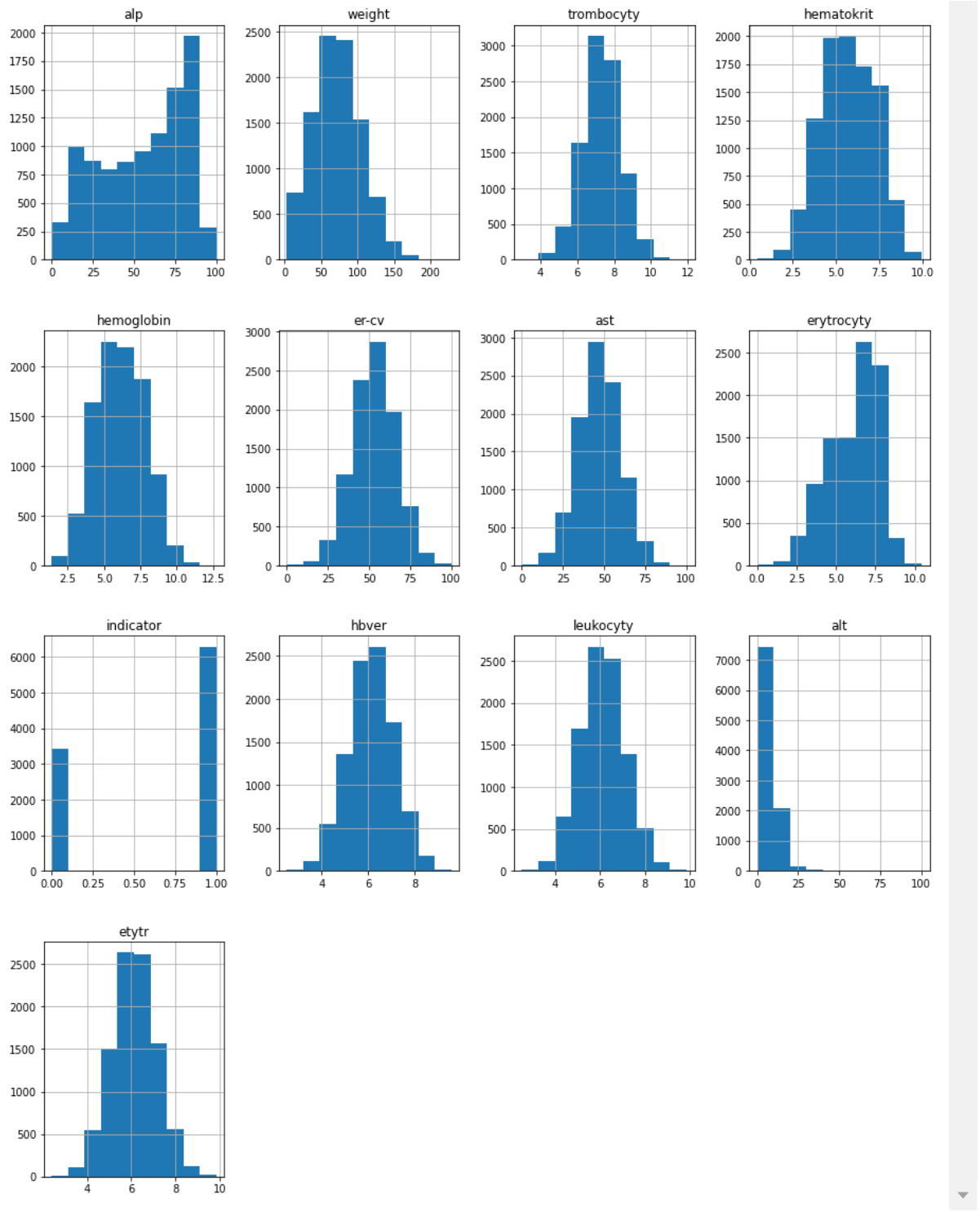
In [52]:

```
fig = plt.figure(figsize = (15,20))
ax = fig.gca()
labor.hist(ax = ax)
```

```
C:\Users\pplev\AppData\Local\Temp/ipykernel_9480/2866531527.py:3: UserWarnin
g: To output multiple subplots, the figure containing the passed axes is bei
ng cleared
  labor.hist(ax = ax)
```

Out[52]:

```
array([[<AxesSubplot:title={'center':'alp'}>,
        <AxesSubplot:title={'center':'weight'}>,
        <AxesSubplot:title={'center':'trombocyty'}>,
        <AxesSubplot:title={'center':'hematokrit'}>],
       [<AxesSubplot:title={'center':'hemoglobin'}>,
        <AxesSubplot:title={'center':'er-cv'}>,
        <AxesSubplot:title={'center':'ast'}>,
        <AxesSubplot:title={'center':'erytrocyty'}>],
       [<AxesSubplot:title={'center':'indicator'}>,
        <AxesSubplot:title={'center':'hbver'}>,
        <AxesSubplot:title={'center':'leukocyty'}>,
        <AxesSubplot:title={'center':'alt'}>],
       [<AxesSubplot:title={'center':'etytr'}>, <AxesSubplot:>,
        <AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```



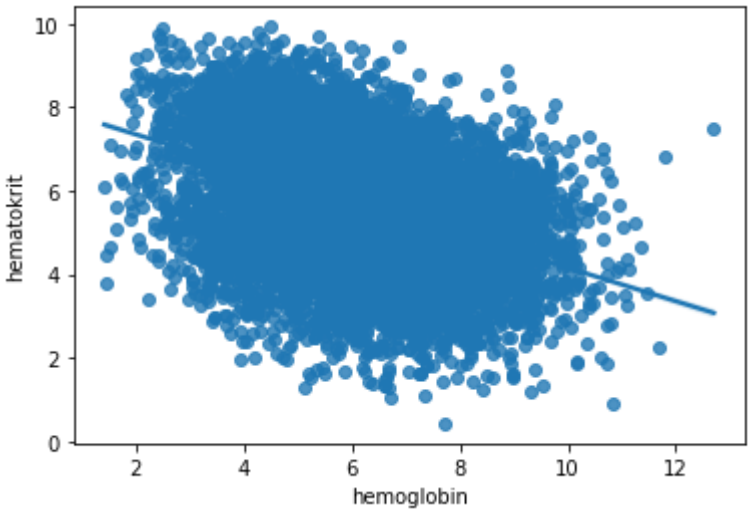
Grafy zobrazujúce počty jednotlivých hodnôt atribútov.

In [53]:

```
sns.regplot(x="hemoglobin",y="hematokrit",data=labor)
```

Out[53]:

```
<AxesSubplot:xlabel='hemoglobin', ylabel='hematokrit'>
```



Tento graf vyjadruje negatívnu koreláciu atribútov hemoglobin a hematokrit. Čo sme vyčítali aj z heatmapy.

In [54]:

```
labor.hematokrit.corr(labor.hemoglobin)
```

Out[54]:

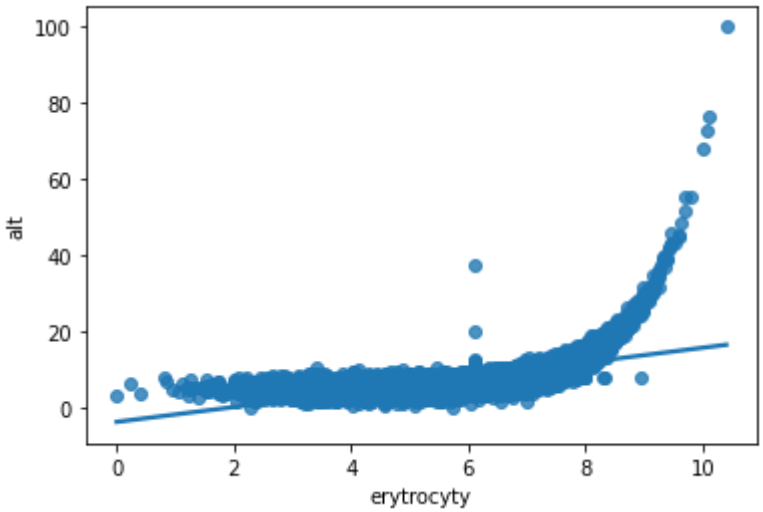
```
-0.42853490576600933
```

In [55]:

```
sns.regplot(x="erytrocyty",y="alt",data=labor)
```

Out[55]:

```
<AxesSubplot:xlabel='erytrocyty', ylabel='alt'>
```



Tento graf vyjadruje najvyššiu koreláciu dvoch atribútov z datasetu labor. Atribúty erytrocyty a alt spolu korelujú. Uvidíme, či nám táto informácia bude v budúcnosti užitočná.

In [56]:

```
labor.alt.corr(labor.erytrocyty)
```

Out[56]:

```
0.6776175252305046
```

In [57]:

```
blood = pd.crosstab(index=labor['indicator'], columns=profiles["blood_group"])
blood.index=["0", "1"]
blood
```

Out[57]:

blood_group	A+	A-	AB+	AB-	B+	B-	O+	O-
0	132	148	129	126	117	116	120	142
1	247	240	237	259	232	246	226	247

Tu máme možnosť vidieť, že krvná skupina nijak neovplyvňuje chorobnosť cukrovky. Môžeme si všimnúť, že pomer chorých ku zdravým je približne 2:1 pre každú krvnú skupinu ako je tomu aj v celom datasete, bez ohľadu na krvnú skupinu.

In [58]:

```
smoker = pd.crosstab(index=labor.smoker, columns=labor.indicator)
smoker.index=["NO","YES"]
smoker
```

Out[58]:

indicator	0.0	1.0
NO	2044	3719
YES	1385	2554

Taktiež ako krvná skupina tak ani to, že či daný pacient fajčí alebo nie nemá signifikantný vplyv na leukémiu. Môžeme vidieť, že pri chorých aj zdravých je percent chorých ako fajčiarov tak aj nefajčiarov.

Formulácia a štatistické overenie hypotéz o dátach

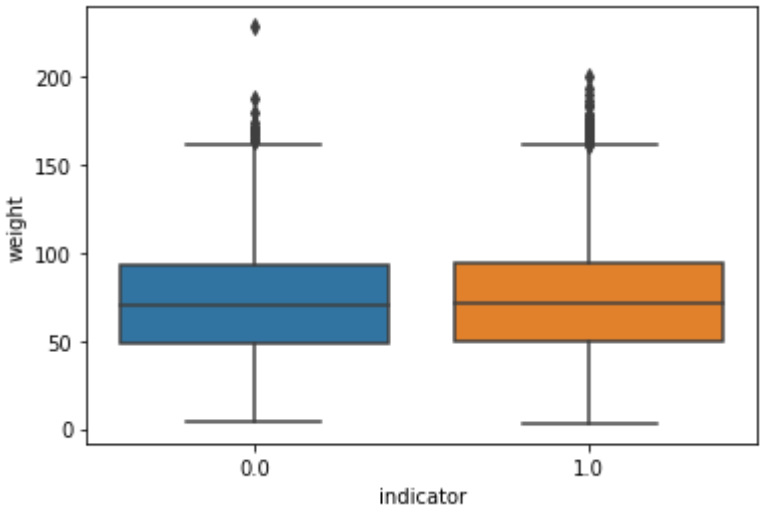
HYPOTÉZA: Váha pacienta ovplyvňuje jeho zdravotný stav.

In [59]:

```
possibly_healthy = labor
sns.boxplot(x = 'indicator', y = 'weight', data=possibly_healthy)
```

Out[59]:

```
<AxesSubplot:xlabel='indicator', ylabel='weight'>
```



Vytiahneme si jednotlivé váhy zdravých a chorých pacientov. Následne si tieto hodnoty zobrazíme na grafoch.

In [60]:

```
healthy = possibly_healthy[possibly_healthy['indicator']==0].weight
ill = possibly_healthy[possibly_healthy['indicator']==1].weight
```

In [61]:

```
labor["indicator"].value_counts()
```

Out[61]:

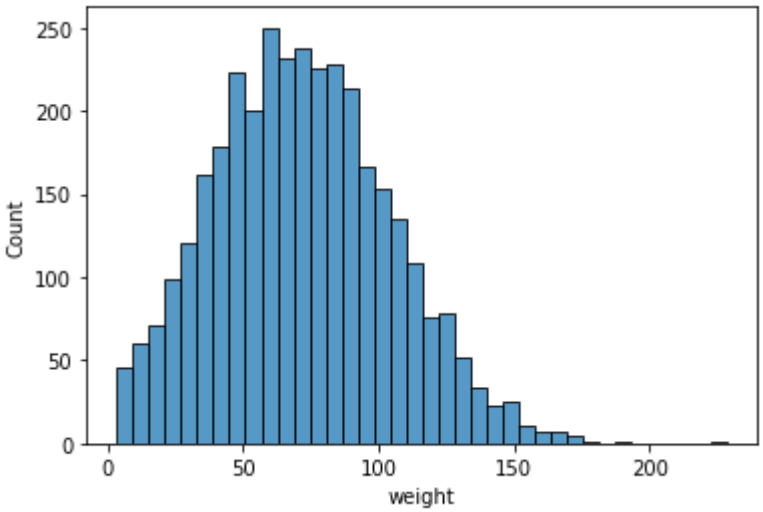
```
1.0    6273
0.0    3429
Name: indicator, dtype: int64
```

In [62]:

```
sns.histplot(healthy)
```

Out[62]:

```
<AxesSubplot:xlabel='weight', ylabel='Count'>
```

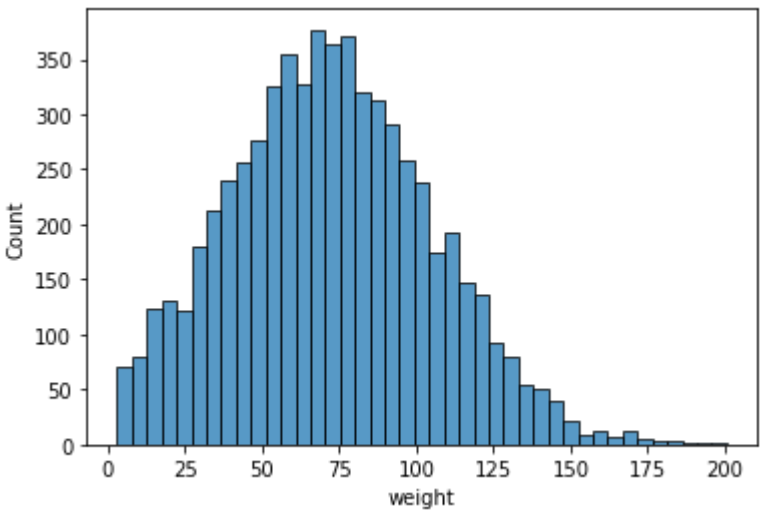


In [63]:

```
sns.histplot(ill)
```

Out[63]:

```
<AxesSubplot:xlabel='weight', ylabel='Count'>
```



Z grafov pravdepodobnostného rozdelenia i boxplotu vidíme, že vo zvolených dátach máme veľa outlierov, preto sme sa ich na dokázanie tejto hypotézy rozhodli vynechať.

In [64]:

```
healthy = healthy[healthy.between(healthy.quantile(.15), healthy.quantile(.85))]
ill = ill[ill.between(ill.quantile(.15), ill.quantile(.85))]
```

In [65]:

```
healthy.describe()
```

Out[65]:

```
count    2399.000000
mean       70.746826
std        18.606322
min        37.460900
25%        55.245460
50%        70.579950
75%        85.980165
max       106.177200
Name: weight, dtype: float64
```

In [66]:

```
ill.describe()
```

Out[66]:

```
count    4391.000000
mean       71.459872
std        18.604570
min        37.294190
25%        56.212840
50%        71.109580
75%        86.530010
max       106.971860
Name: weight, dtype: float64
```

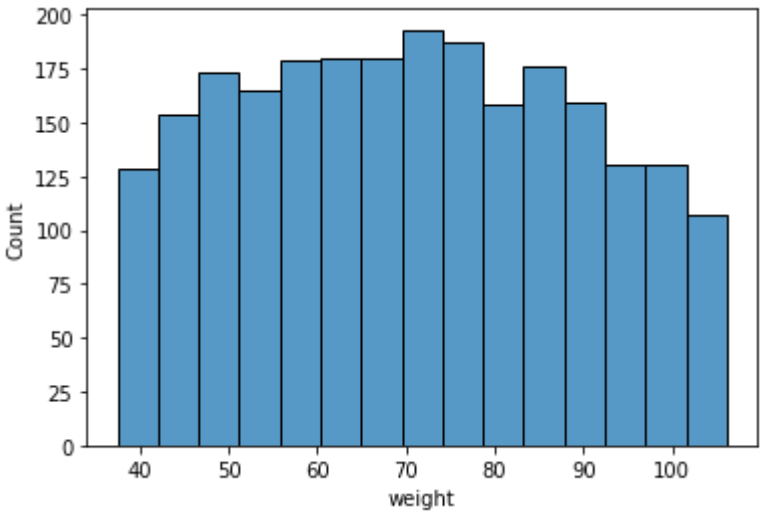
Po odstránení outlierov z dát vieme zachytiť o niečo presnejšie výsledky.

In [67]:

```
sns.histplot(healthy)
```

Out[67]:

```
<AxesSubplot:xlabel='weight', ylabel='Count'>
```

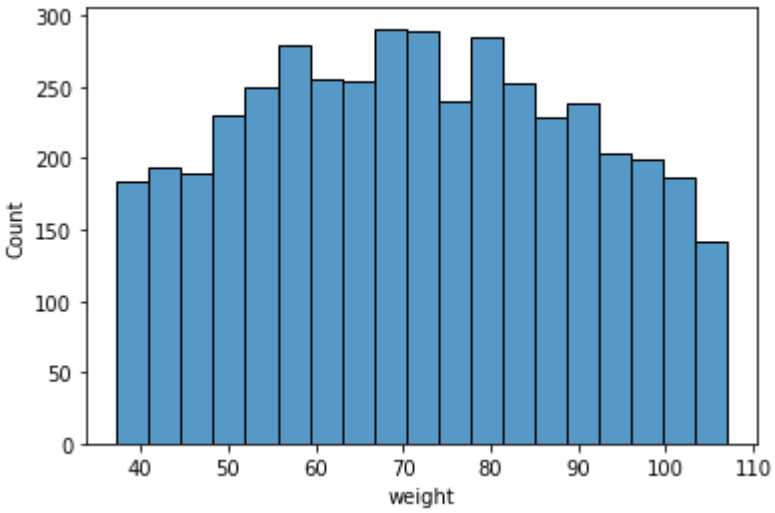


In [68]:

```
sns.histplot(ill)
```

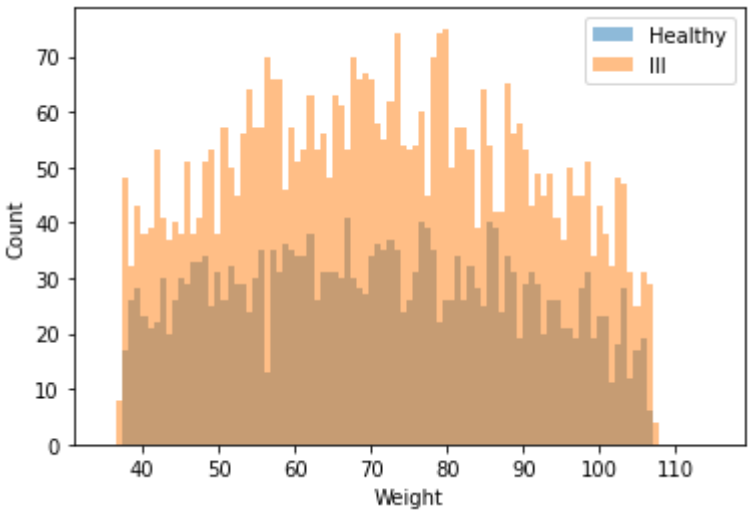
Out[68]:

```
<AxesSubplot:xlabel='weight', ylabel='Count'>
```



In [69]:

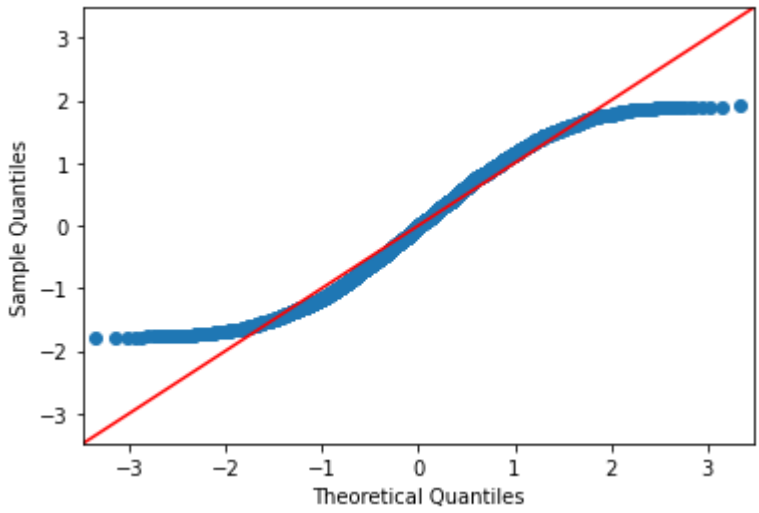
```
bins = np.linspace(35, 115, 100)
pyplot.hist(healthy, bins, alpha=0.5, label='Healthy')
pyplot.hist(ill, bins, alpha=0.5, label='Ill')
pyplot.xlabel('Weight')
pyplot.ylabel('Count')
pyplot.legend(loc='upper right')
pyplot.show()
```



Na grafoch môžme vidieť rozdelenie hodnôt po odstránení outlierov.

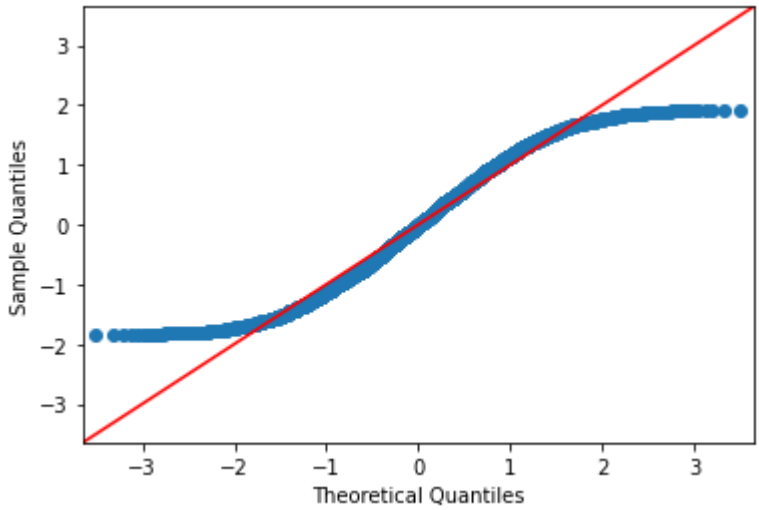
In [70]:

```
_ = sm.ProbPlot(healthy,fit=True).qqplot(line='45')
```



In [71]:

```
_=sm.ProbPlot(ill,fit=True).qqplot(line='45')
```



QQ plot nám určuje, že obe dátové sady pochádzajú z rovnakého rozdelenia. Musíme však overiť normálnosť, teda z akého rozdelenia pochádzajú dáta. Či dáta pochádzajú z normálového rozdelenia zistíme pomocou Shapirovho štatistického testu.

In [72]:

```
stats.shapiro(healthy)
```

Out[72]:

```
ShapiroResult(statistic=0.9671977758407593, pvalue=5.778577786876888e-23)
```

In [73]:

```
stats.shapiro(ill)
```

Out[73]:

```
ShapiroResult(statistic=0.9705511927604675, pvalue=3.9498506024917866e-29)
```

Kedže p-hodnota oboch vzoriek je menšia ako 0.05, nulovú hypotézu zamietame a môžme považovať, že dáta pravdepodobne pochádzajú z iného ako normálneho rozdelenia. Tým pádom môžeme povedať, že predpoklady na t test neboli splnené, preto použijeme mannwhitneyov test kde zistíme,či rozdiel medzi týmito dvoma vzorkami dát je alebo nie je signifikantný.

In [74]:

```
stats.mannwhitneyu(healthy, ill)
```

Out[74]:

```
MannwhitneyuResult(statistic=5153114.0, pvalue=0.14019383650257894)
```

Man-Whiteneyho test nám vrátil pravdepodobnosť chyby, ktorá je väčšia ako 0.10. Môžme teda povedať, že vzorky pocháchdzajú z rovnakého rozdelenia a že nulová hypotéza H0 sa nezamieta. Teda existuje rozdiel medzi váhami chorých a zdravých pacientov ale tento rozdiel nie je signifikantný ale minimálny a zanedbateľný.

In [75]:

```
sms.DescrStatsW(healthy).tconfint_mean()
```

Out[75]:

```
(70.00190018577578, 71.4917508771671)
```

In [76]:

```
sms.DescrStatsW(ill).tconfint_mean()
```

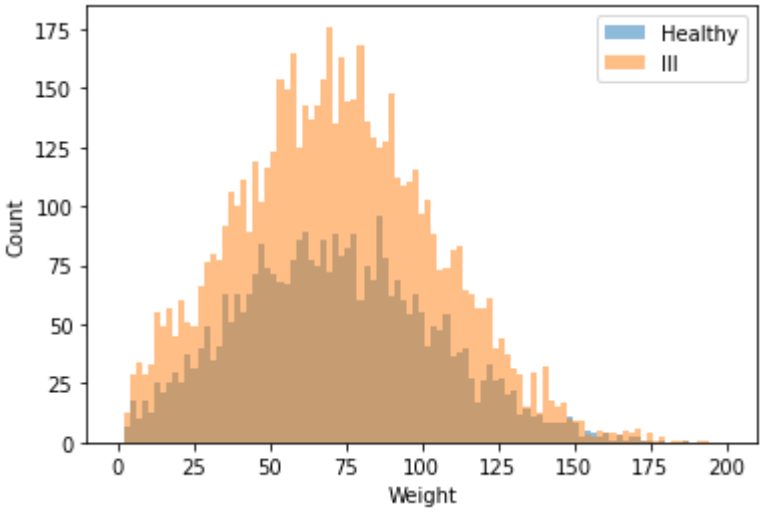
Out[76]:

```
(70.90943704518087, 72.01030636178795)
```

Takto vyzerajú vážené priemery hodnôt váh jednotlivých vzoriek.

In [77]:

```
all_healthy=labor[labor['indicator']==0].weight
all_ill=labor[labor['indicator']==1].weight
bins = np.linspace(0, 200, 100)
pyplot.hist(all_healthy, bins, alpha=0.5, label='Healthy')
pyplot.hist(all_ill, bins, alpha=0.5, label='Ill')
pyplot.xlabel('Weight')
pyplot.ylabel('Count')
pyplot.legend(loc='upper right')
pyplot.show()
```



Z grafu je vidieť, že je viacej chorých s väčšou váhou, ale vzhľadom na to, že pomer chorých a zdravých je 2 ku 1 tak môžeme povedať, že váha zásadne neovplyvňuje to, či pacient má alebo nemá leukémiu.

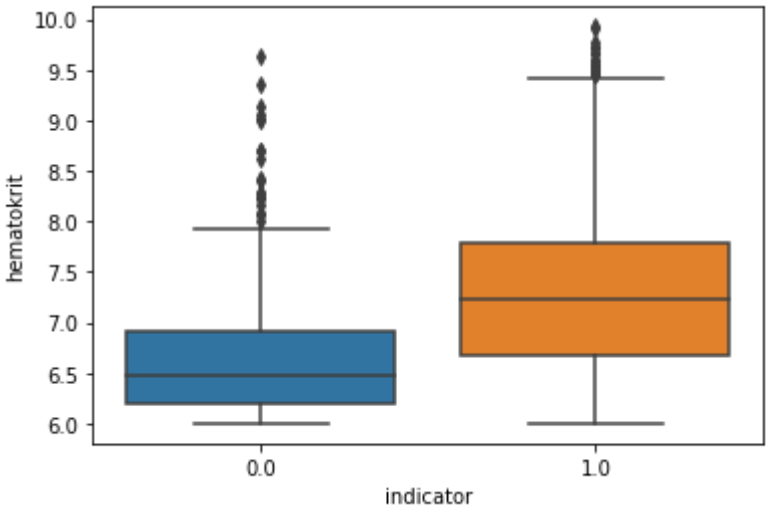
HYPOTÉZA: Ak je hodnota atribútu hematokrit viac ako 6, pacient je pravdepodobne chorý.

In [78]:

```
possibly_healthy = labor[(labor.hematokrit >6)]
sns.boxplot(x = 'indicator', y = 'hematokrit', data=possibly_healthy)
```

Out[78]:

```
<AxesSubplot:xlabel='indicator', ylabel='hematokrit'>
```



In [79]:

```
possibly_healthy["indicator"].value_counts()
```

Out[79]:

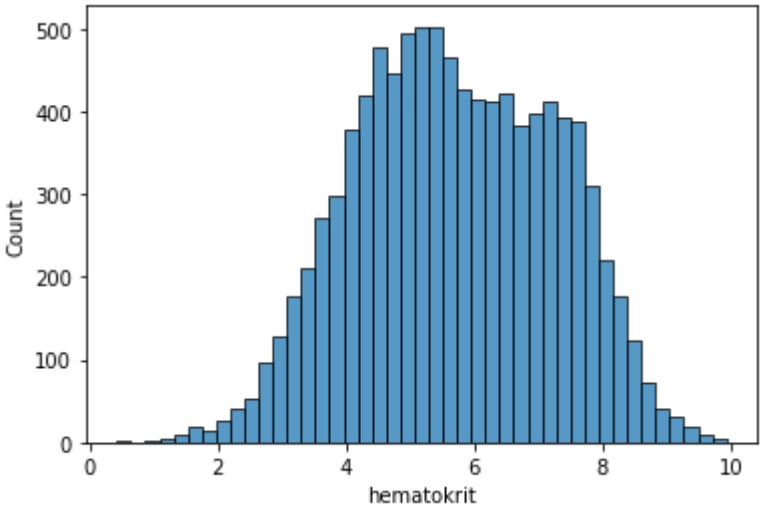
```
1.0    3533
0.0     603
Name: indicator, dtype: int64
```

In [80]:

```
sns.histplot(labor['hematokrit'])
```

Out[80]:

```
<AxesSubplot:xlabel='hematokrit', ylabel='Count'>
```



In [81]:

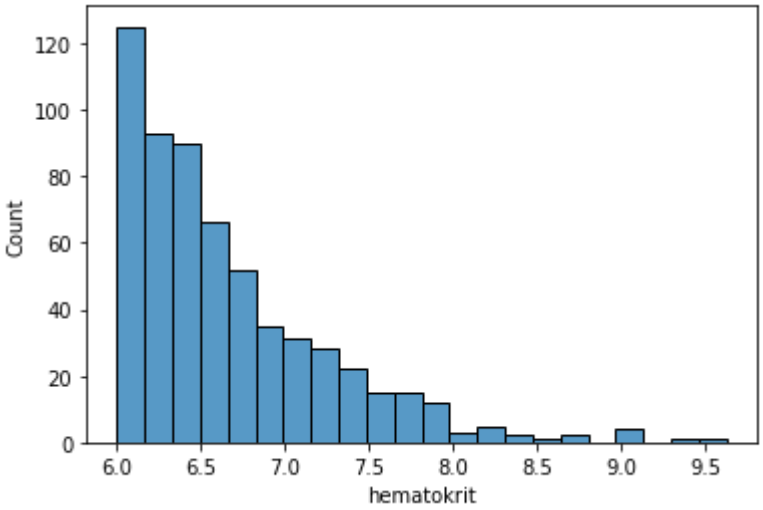
```
healthy = possibly_healthy[possibly_healthy['indicator']==0].hematokrit
ill = possibly_healthy[possibly_healthy['indicator']==1].hematokrit
```

In [82]:

```
sns.histplot(healthy)
```

Out[82]:

```
<AxesSubplot:xlabel='hematokrit', ylabel='Count'>
```

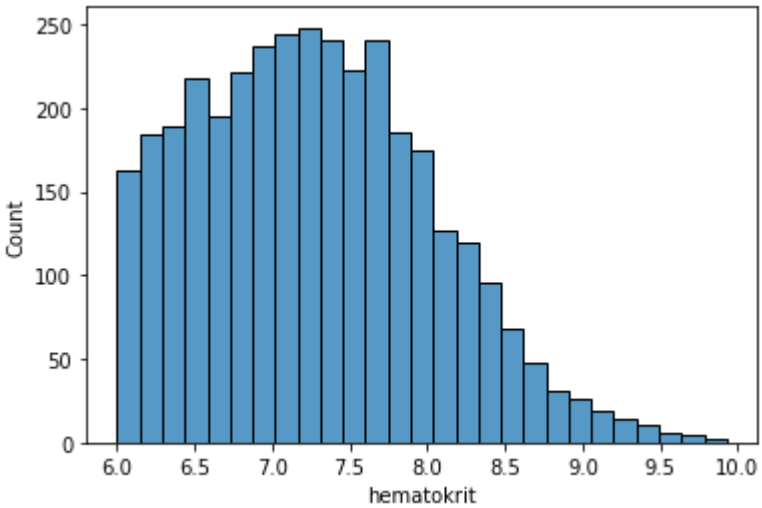


In [83]:

```
sns.histplot(ill)
```

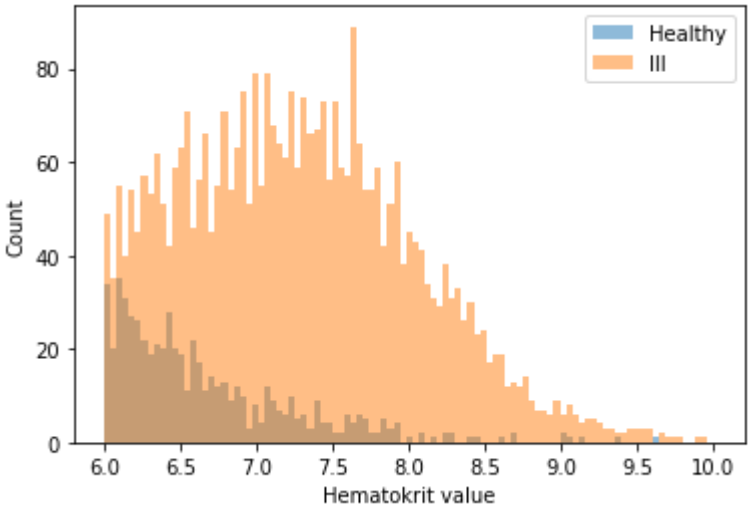
Out[83]:

```
<AxesSubplot:xlabel='hematokrit', ylabel='Count'>
```



In [84]:

```
bins = np.linspace(6, 10, 100)
pyplot.hist(healthy, bins, alpha=0.5, label='Healthy')
pyplot.hist(ill, bins, alpha=0.5, label='Ill')
pyplot.legend(loc='upper right')
pyplot.xlabel('Hematokrit value')
pyplot.ylabel('Count')
pyplot.show()
```



Z grafov pravdepodobnostného rozdelenia i boxplotu vidíme, že vo zvolených dátach máme veľa outlierov, preto sme sa ich na dokázanie tejto hypotézy rozhodli vynechať.

In [85]:

```
healthy = healthy[healthy.between(healthy.quantile(.15), healthy.quantile(.85))] # without
ill = ill[ill.between(ill.quantile(.15), ill.quantile(.85))] # without outliers
```

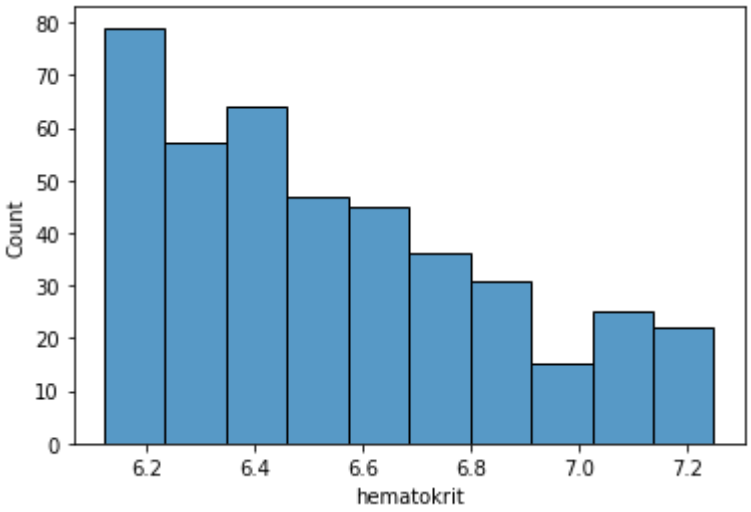
Po odstránení outlierov z dát vieme zachytiť o niečo presnejšie výsledky.

In [86]:

```
sns.histplot(healthy)
```

Out[86]:

```
<AxesSubplot:xlabel='hematokrit', ylabel='Count'>
```

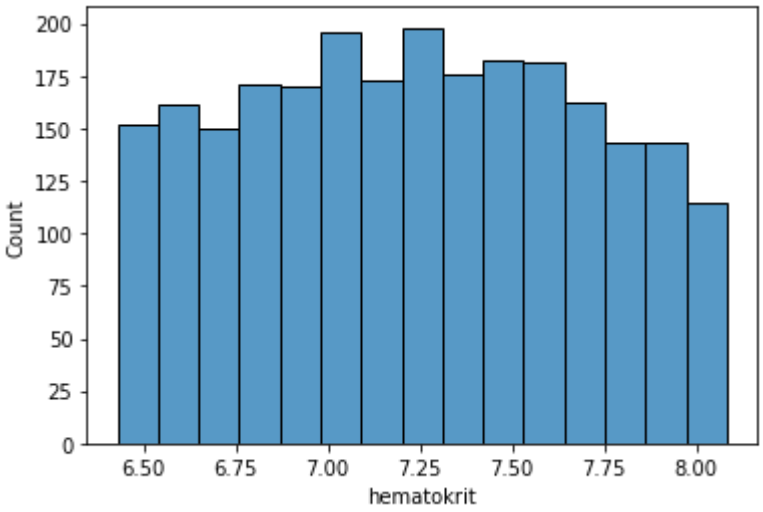


In [87]:

```
sns.histplot(ill)
```

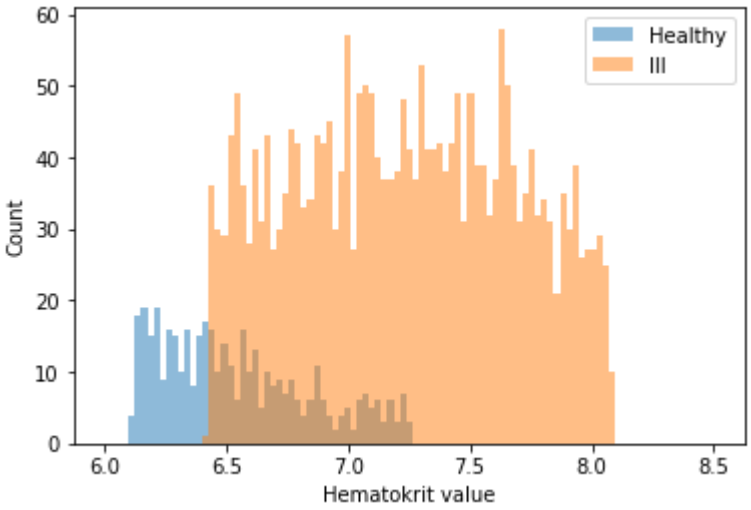
Out[87]:

```
<AxesSubplot:xlabel='hematokrit', ylabel='Count'>
```



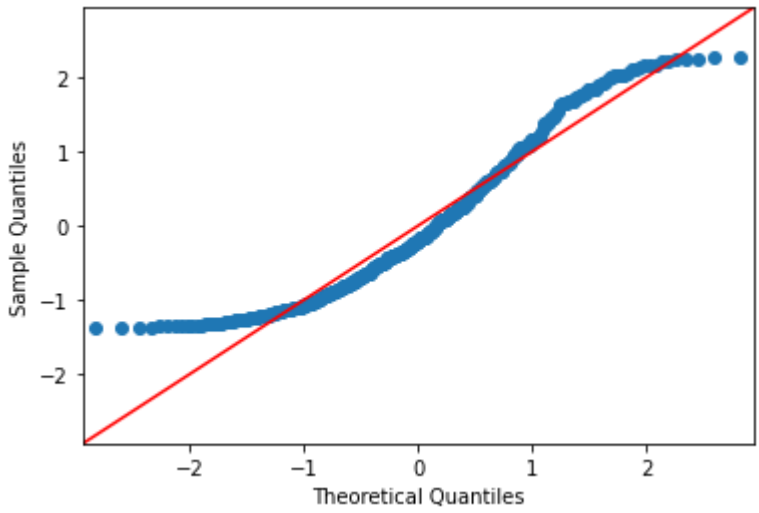
In [88]:

```
bins = np.linspace(6, 8.5, 100)
pyplot.hist(healthy, bins, alpha=0.5, label='Healthy')
pyplot.hist(ill, bins, alpha=0.5, label='Ill')
pyplot.legend(loc='upper right')
pyplot.xlabel('Hematokrit value')
pyplot.ylabel('Count')
pyplot.show()
```



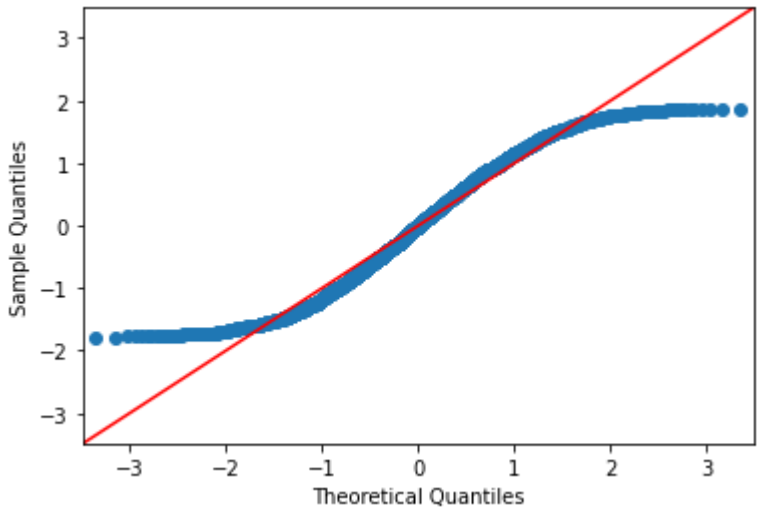
In [89]:

```
_ = sm.ProbPlot(healthy,fit=True).qqplot(line='45')
```



In [90]:

```
_ = sm.ProbPlot(ill,fit=True).qqplot(line='45')
```



QQ plot nám určuje, že obe dátové sady pochádzajú z rovnakého rozdelenia. Musíme však overiť normálnosť, teda z akého rozdelenia pochádzajú dáta. Či dáta pochádzajú z normálového rozdelenia zistíme pomocou Shapirovho štatistického testu.

In [91]:

```
stats.shapiro(healthy)
```

Out[91]:

```
ShapiroResult(statistic=0.9365670680999756, pvalue=2.090870662371791e-12)
```

In [92]:

```
stats.shapiro(ill)
```

Out[92]:

```
ShapiroResult(statistic=0.965526819229126, pvalue=7.179463330322595e-24)
```

Rovnako ako pri prvej hypotéze p-hodnota oboch vzoriek je menšia ako 0.05, čo znamená, že nulovú hypotézu zamietame. Preto použijeme mannwhitneyov test kde zistíme,či rozdiel medzi týmito dvoma vzorkami dát je alebo nie je signifikantný.

In [93]:

```
stats.mannwhitneyu(healthy, ill)
```

Out[93]:

```
MannwhitneyuResult(statistic=115540.0, pvalue=4.6860513610310735e-144)
```

Keďže p value je <0,001 pravdepodobnosť chyby je menej ako 1 promile. Rozdiel v hodnotách hematokrit medzi chorými a zdravými ľudmi je štatisticky viditeľný teda signifikantný.

In [94]:

```
sms.DescrStatsW(healthy).tconfint_mean()
```

Out[94]:

```
(6.516524224485295, 6.575666274327055)
```

In [95]:

```
sms.DescrStatsW(ill).tconfint_mean()
```

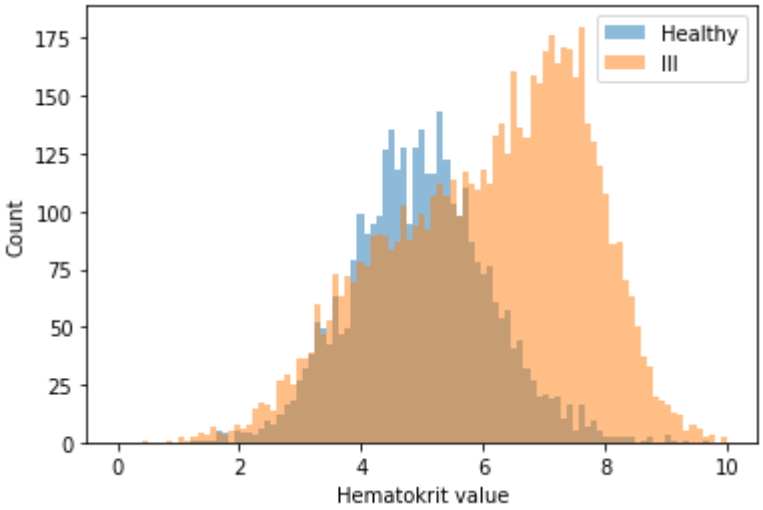
Out[95]:

```
(7.217278550202204, 7.252987426344504)
```

Takto vyzerajú vážené priemery jednotlivých vzoriek hodnôt hematokritov.

In [96]:

```
all_healthy=labor[labor['indicator']==0].hematokrit
all_ill=labor[labor['indicator']==1].hematokrit
bins = np.linspace(0, 10, 100)
pyplot.hist(all_healthy, bins, alpha=0.5, label='Healthy')
pyplot.hist(all_ill, bins, alpha=0.5, label='Ill')
pyplot.legend(loc='upper right')
pyplot.xlabel('Hematokrit value')
pyplot.ylabel('Count')
pyplot.show()
```



Dokázali sme teda, že naozaj môžeme vidieť, že väčšina ľudí, s hodnotou hematokritu nad 6 je chorých.

ZÁVER

Výsledky:

- Niektoré dáta mali nevhodný formát, preto sme ich na začiatku museli upraviť. Pokiaľ bol počet prázdnych numerických hodnôt malý, vhodnou alternatívou bolo nahradenie hodnôt priemerom ostatných prítomných hodnôt. Ostatné nevhodné zaznamenané hodnoty atribútov sme buď opravili na jednotný formát alebo opravili chyby.
- Niektoré atribúty ako napríklad weight, ktorý reprezentuje váhu pacienta mali nezmyselmé hodnoty tak bolo nutné tieto hodnoty odstrániť.
- V datasete sa nachádza viacero závislých premenných ale aj takých, ktoré majú negatívnu koreláciu s ostatnými atribútmi.
- Párovú analýzu údajov sme vykonávali nad viacerými dvojicami atribútov.

- Podarilo sa nám skúmať dáta a vykonávať štatistické testy na základe nami stanovených hypotéz. Na základe tejto analýzy a testov sme vyvodili horespomenuté výsledky a závery.
- Prípadne ďalšie nezrovnalosti v datasete budeme riešiť v ďalších fázach projektu.

Spojenie oboch tabuliek do jednej

Predpripravili sme si dáta a spojili sme obe tabuľky do jednej. V budúcich fázach sa nám to možno bude hodiť.

In [97]:

```
merged = pd.merge(labor, profiles, on=["ssn", "name"])
```

In [98]:

```
pd.set_option("display.max_rows", None, "display.max_columns", None)
print(profiles)
```

```
                                            residence          ssn  \
0     55180 Charlotte Mission\nPort Janiceton, MD 01615  707-91-3436
1     4157 Chelsea Extension Apt. 138\nPhillipstown,...  882-73-6960
2               8890 Rogers Trail\nNew April, VT 56293  395-27-1265
3     6073 Roger Via Suite 739\nPort Johnfort, NM 79606  708-36-7168
4     6685 Jason Trafficway Apt. 492\nWest Deantown,...  183-78-8749
5      55098 Timothy Mall\nLake Rhondachester, ND 95966  290-30-8274
6               13209 Carol Grove\nPaulaview, LA 80082  335-75-2783
7          7897 Joseph Valleys\nNelsonchester, KS 39327  147-66-1480
8     070 Brandi Wells Suite 668\nWest Matthew, CA 5...  254-97-8945
9                          USS Gonzalez\nFPO AA 45598  579-57-3686
10       543 Malone Plain Apt. 241\nDixonbury, NE 86064  278-44-3737
11                   Unit 8640 Box 0927\nDPO AA 01035  543-07-8565
12              2170 Jacobs Light\nGeorgeton, MN 33093  200-20-4285
13        2259 Roy Circle Suite 469\nRiceland, LA 03407  063-47-0942
14    0080 Stephanie Stravenue\nLake Vincentbury, MN...  394-55-7199
15              928 Taylor Road\nSouth Javier, AL 48046  439-78-2049
16              554 Angela Centers\nCraigland, WA 49091  403-88-9976
17         08090 Knox Turnpike\nSouth Melissa, MT 46137  225-27-1637
```

In [99]:

```
pd.set_option("display.max_rows", None, "display.max_columns", None)
print(labor)
```

```
      relationship smoker         alp     weight  trombocyty  hematokrit
\
0         divorced    yes   69.297540   96.36107    6.160090    5.430570
1        separated    yes   77.145640   87.14201    7.566480    5.491490
2           single     no   80.333350   77.58017    6.401810    3.432400
3         divorced     no   87.096550  112.49541    6.509800    4.315590
4              nop    yes   68.787240   85.92167    8.611020    7.984290
5           single     no   79.708620  129.95085    7.367040    5.195240
6           single     no   10.003150   87.46249    8.978060    8.215040
7           single    yes   70.735270   77.14382    8.039980    4.130440
8          married    yes   59.642350   78.69870    6.796710    6.353650
9          married     no   28.872070   61.87194    7.203420    7.892210
10         married    yes   61.581130  139.22602    8.245600    7.787920
11             nop     no   16.156060  100.64335    7.484450    7.814590
12       separated    yes   12.182920   18.82065    6.992360    6.769050
13         married    yes   69.137790   28.44820    7.191920    5.367200
14             nop     no   62.741910   99.61714    8.001600    5.735700
15        divorced     no   78.918530   65.07001    7.557410    7.133900
16             nop     no   21.419620   89.18042    5.068790    6.871180
```