

# IAU Projekt

## Projekt vypracovali:

- Peter Smreček - 50%
- Anetta Langová - 50%

## Zadanie (the quest)

- Každá dvojica bude pracovať s pridelenou dátovou sadou (3. týždeň).
- Vašou úlohou je vedieť predikovať závislú hodnotu (indikator)

Budete sa musieť pritom vysporiadať s viacerými problémami, ktoré sa v dátach nachádzajú ako formáty dát, chýbajúce, vychýlené hodnoty a pod.

```
In [1]:  
import matplotlib.pyplot as plt  
import seaborn as sns  
import pandas as pd  
import numpy as np  
import scipy.stats as stats  
import statsmodels.api as sm  
import statsmodels.stats.api as sms  
import statsmodels.stats as sms_stats
```

```
In [2]:  
filename_labor = "046/labor.csv"  
labor = pd.read_csv(filename_labor, sep='\t')  
labor.head()
```

	Unnamed: 0	weight	ssn	hemoglobin	alp	etytr	alt	erytrocyt	hbver	na
0	0	12.35740	803- 27- 3974	5.94182	86.80991	7.12559	2.18482	7.53345	7.51373	Ka Rodrigi
1	1	118.10209	205- 46- 9278	6.45407	79.22919	5.34025	1.60706	6.53048	7.30977	Di +
2	2	89.97897	507- 12- 0831	9.73090	17.97254	9.49744	1.75153	5.96430	8.02289	Da Bry I
3	3	137.89307	328- 79- 8098	8.65753	19.63713	8.90814	5.80869	8.47758	7.37768	Jos Hickm
4	4	95.55653	307- 37- 5739	5.96644	82.63100	6.84092	3.16471	7.38053	8.07490	Fra Glo

```
In [3]:  
filename_profiles = "046/profiles.csv"  
profiles = pd.read_csv(filename_profiles, sep='\t')  
profiles.head()
```

Out[3]:

	Unnamed: 0	race	residence	job	birthdate	company	blood_group	sex
0	0	White	108 Pham Loaf\nNew Shelby, IN 31526	Magazine features editor	2007/09/11	Reynolds, Stewart and Tanner	O+	F
1	1	Asian	2246 Tammy Cliffs Apt. 057\nNorth Kim, MI 55878	Investment banker, operational	1971/05/20	Mcdonald-White	A-	F
2	2	White	12245 Maxwell Island\nNorth Benjamin, KY 38697	Charity fundraiser	09 Oct 1945	Waters, Davis and Mcintyre	AB+	F
3	3	White	17000 Saunders Circles Apt. 457\nDonaldhaven, ...	Engineer, production	04/25/2006, 00:00:00	Watson and Sons	A+	F
4	4	Hawaiian	43712 Andrea Expressway\nNorth Mckenzie, AZ 31049	IT consultant	1936-03-01	Alvarez PLC	B+	M

## 1. Základný opis dát spolu s ich charakteristikami (5 bodov)

- Pre dosiahnutie plného počtu bodov uvedťe
- počet záznamov,
- počet atribútov,
- ich typy,
- pre zvolené významné atribúty ich distribúcie, základné deskriptívne štatistiky a pod.

## DATASET labor.csv

Dataset labor obsahuje záznamy z vyšetrení konkrétnych osôb identifikovaných na základe SSN. Záznamy z vyšetrení pozostavávajú z niekoľkých výsledkov rôznych krvných testov.

In [4]:

```
labor.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10002 entries, 0 to 10001
Data columns (total 18 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Unnamed: 0        10002 non-null   int64  
 1   weight            10002 non-null   float64 
 2   ssn               10002 non-null   object  
 3   hemoglobin       9972 non-null   float64 
 4   alp               9972 non-null   float64 
 5   etytr             9972 non-null   float64 
 6   alt               9971 non-null   float64 
 7   erytrocyty       9972 non-null   float64 
 8   hbver             9972 non-null   float64 
 9   name              10002 non-null   object  
 10  hematokrit       9972 non-null   float64
```

```
11 indicator      10002 non-null  float64
12 er-cv          9972 non-null  float64
13 leukocyty     9972 non-null  float64
14 smoker         10002 non-null  object
15 relationship   10002 non-null  object
16 ast            9972 non-null  float64
17 trombocyty    9972 non-null  float64
dtypes: float64(13), int64(1), object(4)
memory usage: 1.4+ MB
```

Pomocou tohto výpisu sa dajú zistíť všetky potrebné informácie:

- Počet záznamov (2. riadok): 10002 (0 - 10001)
- Počet atribútov (3. riadok): 18 (0 - 17)
- Typy atribútov (4. stĺpec v tabuľke): int64, float64, object

Okrem iného sa dá z toho vyčítať, že všetky záznamy sú priradené kontkrétnym osobám.

```
In [5]: labor.duplicated().any()
```

```
Out[5]: False
```

V dátach sa nenachádzajú duplicitné záznamy.

## Kategórické atribúty

Ďalší zaujímavý atribút je SSN. Ide o Social Security Number (SSN), ktoré sa používa podobne ako u nás rodné číslo, čiže ide len o administratívny údaj.

Teraz sa pozrieme na ostatné atribúty typu object a ich hodnoty.

```
In [6]: labor['smoker'].value_counts()
```

```
Out[6]: no      5497
yes     2093
Y       1917
N       495
Name: smoker, dtype: int64
```

Hoci atribút 'smoker' je dopytovací, hodnoty sú štyri. Je pravdepodobné, že ide o označenie tých istých odpovedí rôznym spôsobom, preto môžeme zlúčiť hodnoty 'yes' s 'Y' a 'no' s 'N' a upraviť ich na číselnú hodnotu 1 prípadne 0.

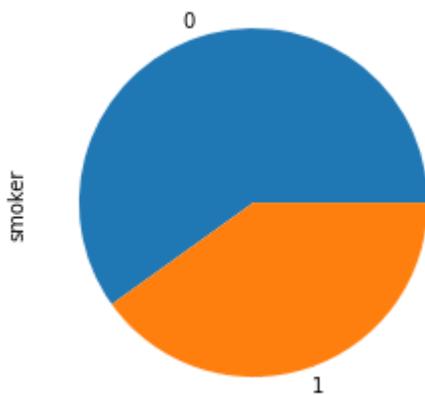
```
In [7]: labor["smoker"].replace({"Y": 1, "N": 0, "yes": 1, "no": 0}, inplace=True)
```

```
In [8]: labor['smoker'].value_counts()
```

```
Out[8]: 0      5992
1      4010
Name: smoker, dtype: int64
```

```
In [9]: labor['smoker'].value_counts().plot(kind='pie')
```

```
Out[9]: <AxesSubplot:ylabel='smoker'>
```



```
In [10]: labor['relationship'].value_counts()
```

```
Out[10]:
```

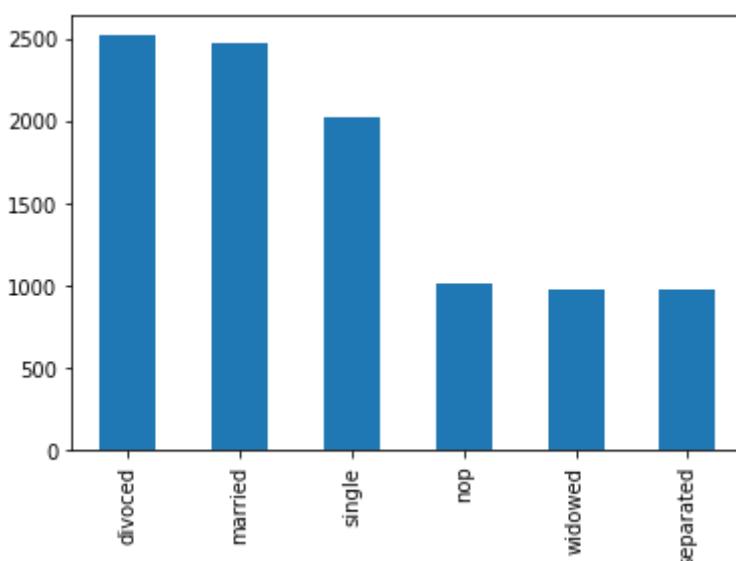
relationship	count
divoced	2523
married	2474
single	2028
nop	1017
widowed	982
separated	978

Name: relationship, dtype: int64

Atribút 'relationship' neobsahuje žiadne NaN hodnoty alebo nejednoznačné typy hodnôt.

```
In [11]: labor['relationship'].value_counts().plot(kind='bar')
```

```
Out[11]: <AxesSubplot:>
```



```
In [12]: labor.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10002 entries, 0 to 10001
Data columns (total 18 columns):
 #   Column      Non-Null Count  Dtype  
 --- 
 0   Unnamed: 0   10002 non-null   int64  
 1   weight      10002 non-null   float64 
 2   ssn         10002 non-null   object  
 3   hemoglobin  9972 non-null   float64 
 4   alp        9972 non-null   float64
```

```

5    etytr          9972 non-null   float64
6    alt            9971 non-null   float64
7  erytrocyty     9972 non-null   float64
8    hbver          9972 non-null   float64
9    name           10002 non-null   object
10 hematokrit     9972 non-null   float64
11 indicator       10002 non-null   float64
12 er-cv           9972 non-null   float64
13 leukocyty      9972 non-null   float64
14 smoker          10002 non-null   int64
15 relationship    10002 non-null   object
16 ast             9972 non-null   float64
17 trombocyty     9972 non-null   float64
dtypes: float64(13), int64(2), object(3)
memory usage: 1.4+ MB

```

### Premenovanie prvého stĺpca:

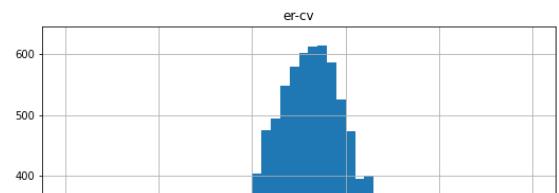
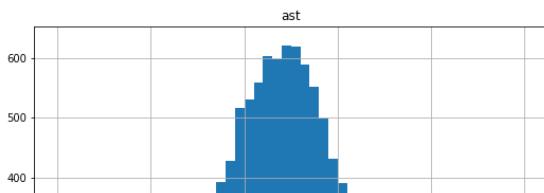
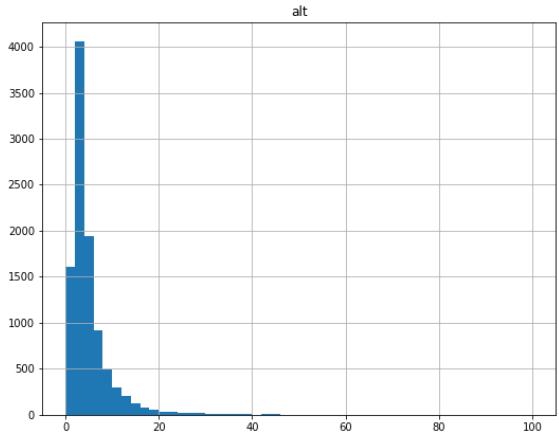
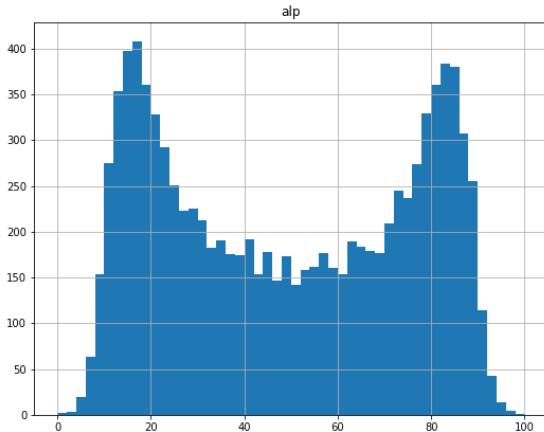
Podľa vypísaných dát je možné vidieť, že úplne prvý nemenovaný atribút je len na označenie, očíslovanie.

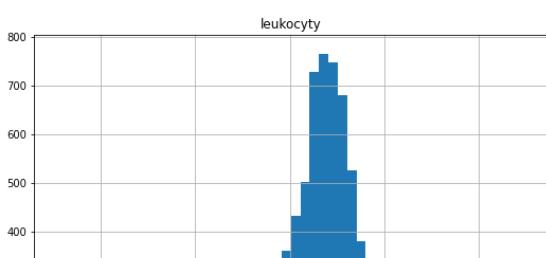
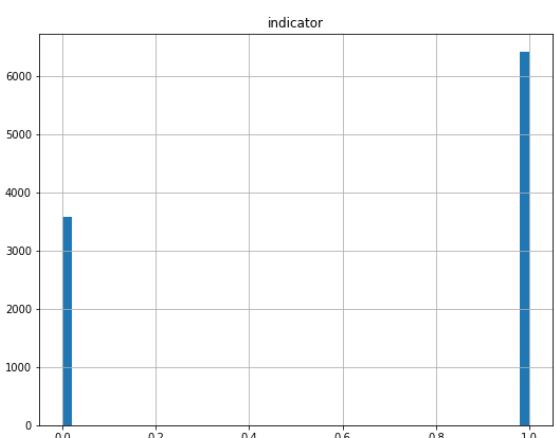
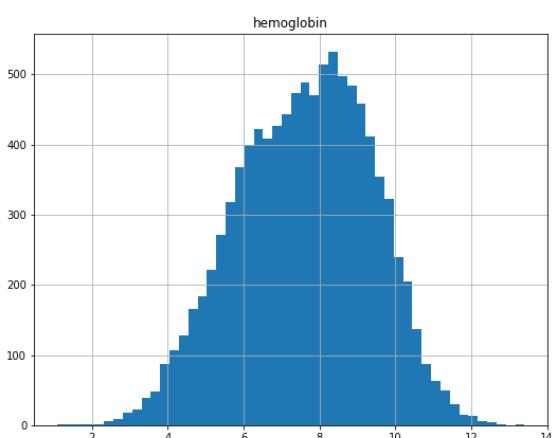
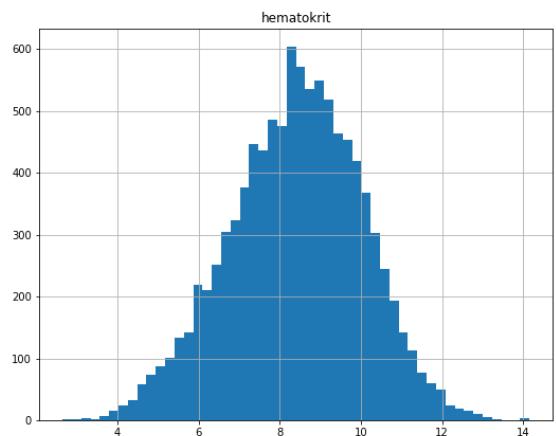
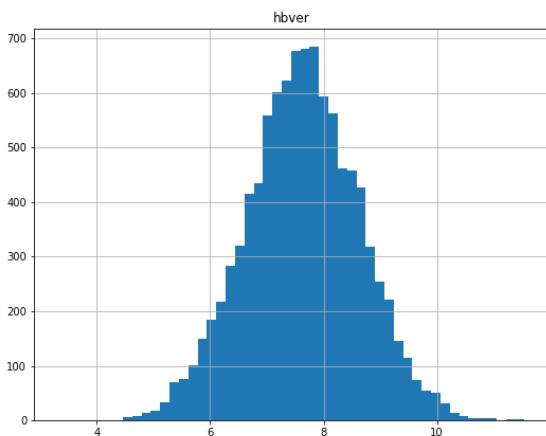
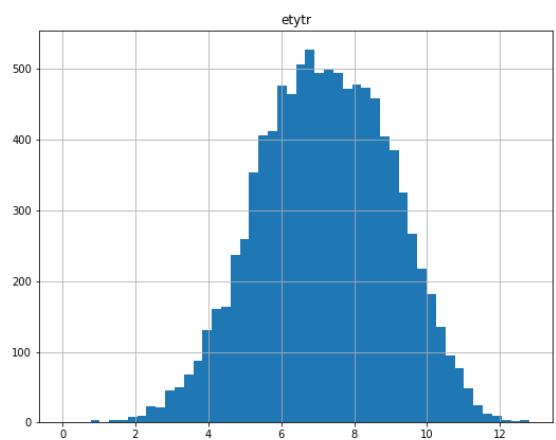
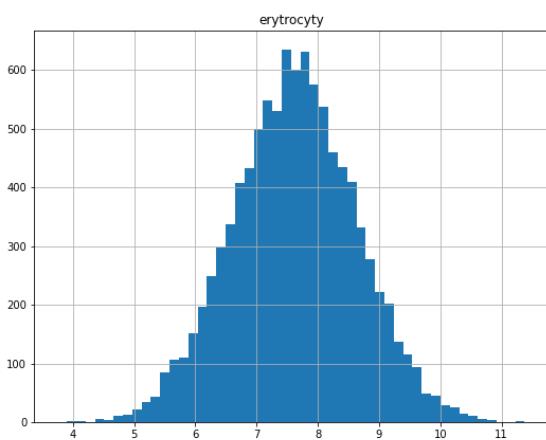
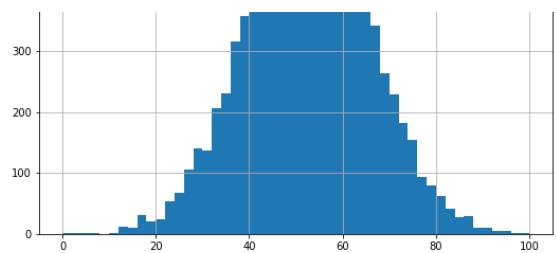
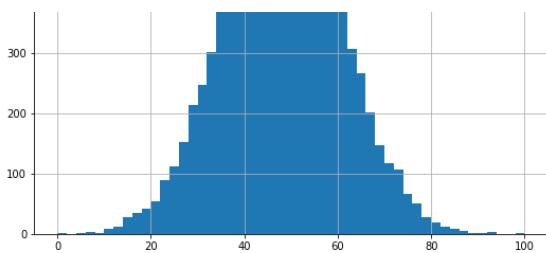
```
In [13]: labor.rename(columns = {"Unnamed: 0": "index"}, inplace = True)
```

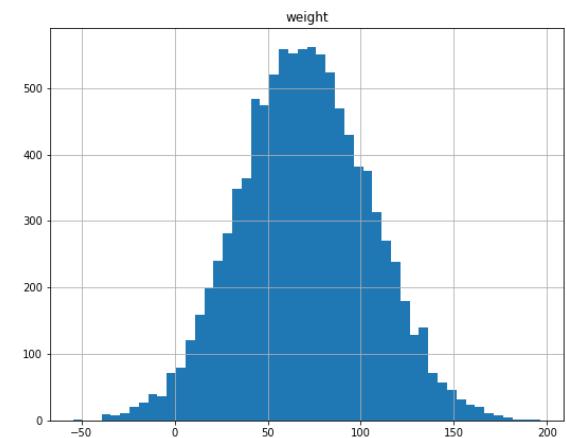
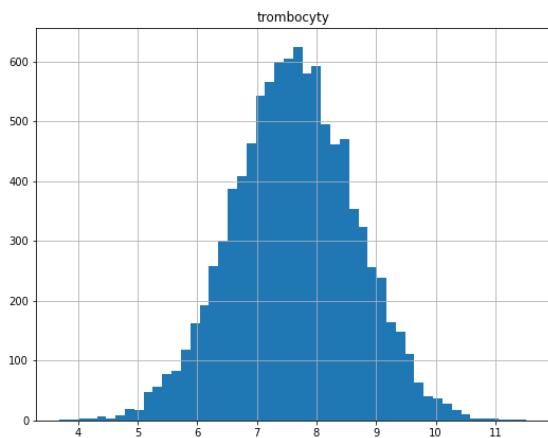
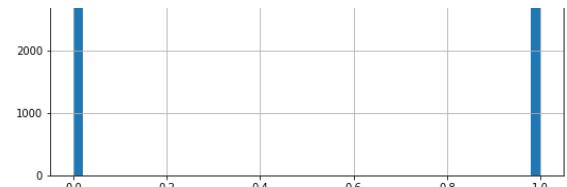
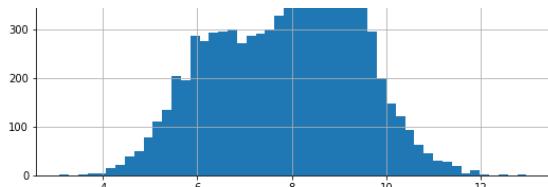
### Numerické artibúty:

```
In [14]: labor.loc[:, labor.columns.difference(["index"])].hist(layout=(7,2), sharex=False, s
```

```
Out[14]: array([[[<AxesSubplot:title={'center':'alp'}>,
                  <AxesSubplot:title={'center':'alt'}>],
                 [<AxesSubplot:title={'center':'ast'}>,
                  <AxesSubplot:title={'center':'er-cv'}>],
                 [<AxesSubplot:title={'center':'erytrocyty'}>,
                  <AxesSubplot:title={'center':'etytr'}>],
                 [<AxesSubplot:title={'center':'hbver'}>,
                  <AxesSubplot:title={'center':'hematokrit'}>],
                 [<AxesSubplot:title={'center':'hemoglobin'}>,
                  <AxesSubplot:title={'center':'indicator'}>],
                 [<AxesSubplot:title={'center':'leukocyty'}>,
                  <AxesSubplot:title={'center':'smoker'}>],
                 [<AxesSubplot:title={'center':'trombocyty'}>,
                  <AxesSubplot:title={'center':'weight'}>]], dtype=object)
```

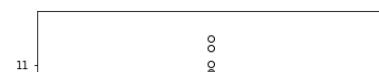
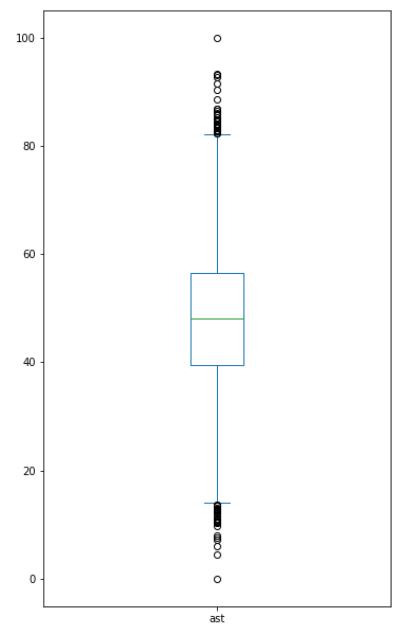
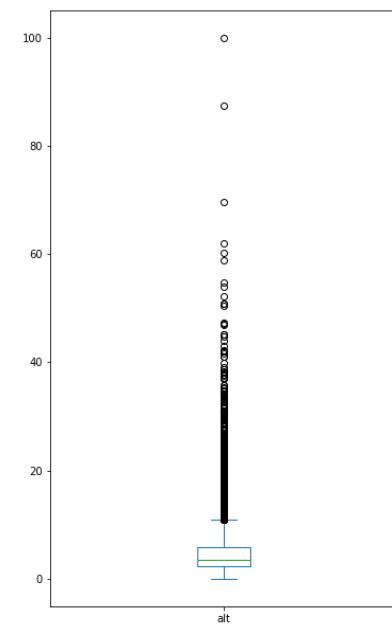
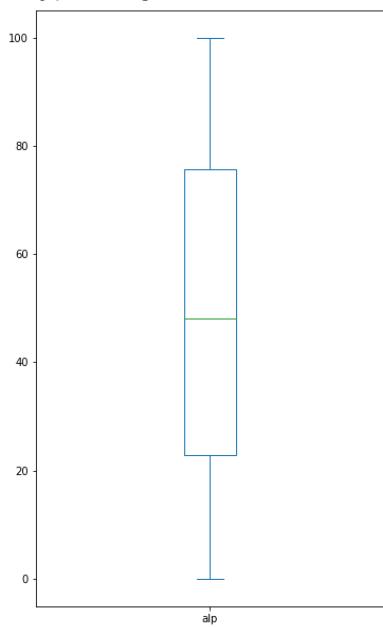


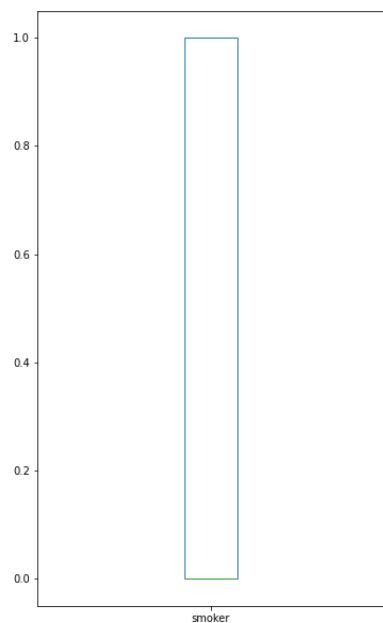
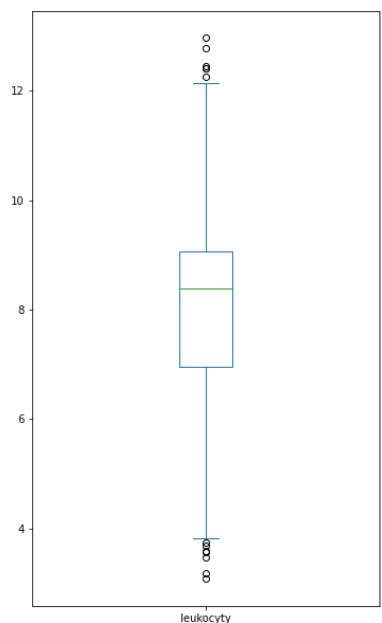
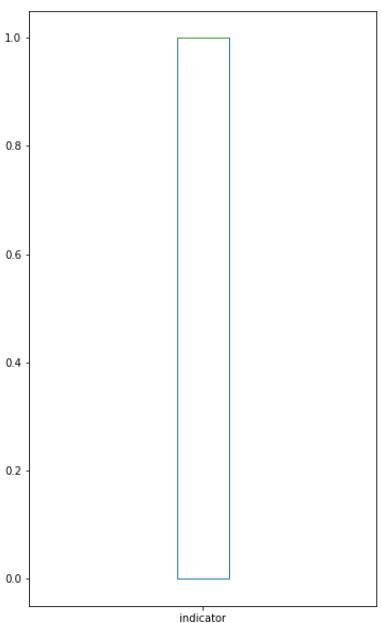
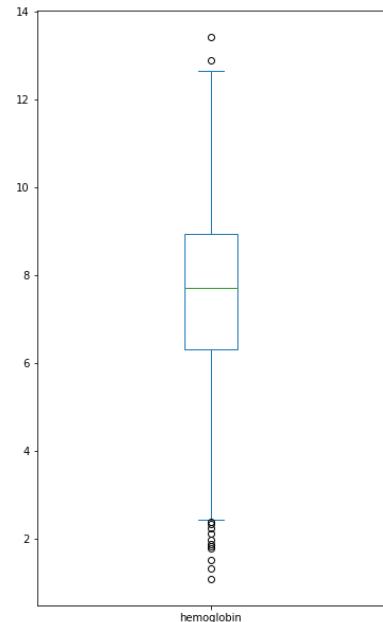
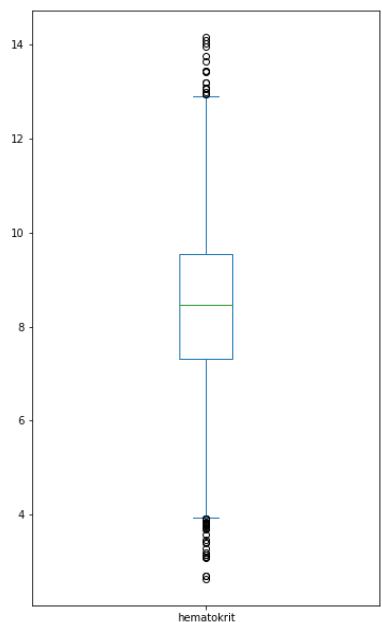
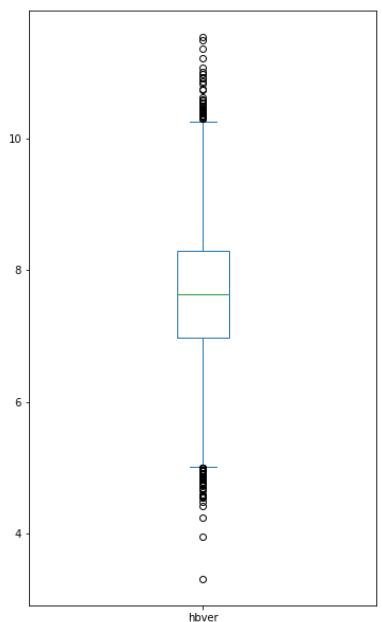
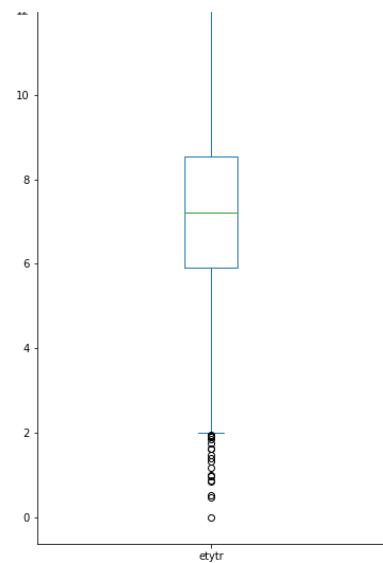
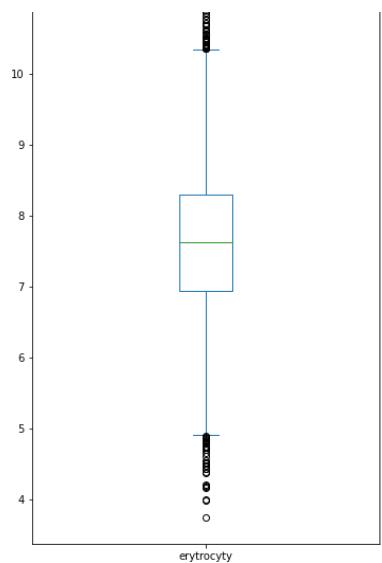
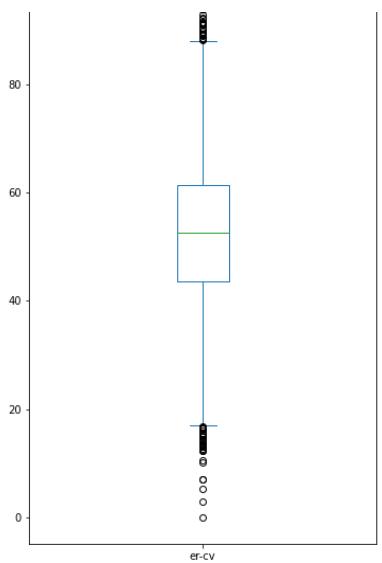


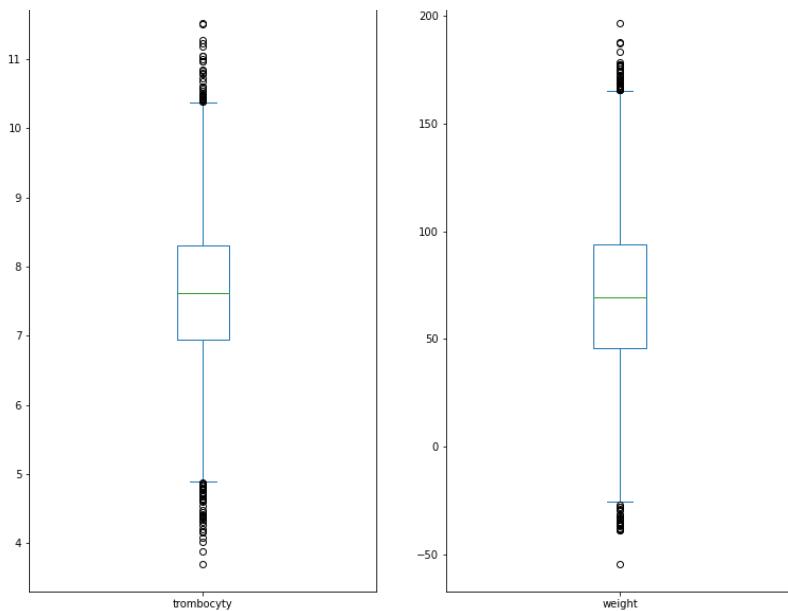


```
In [15]: labor.loc[:, labor.columns.difference(["index"])].plot(kind='box', subplots=True, la
```

```
Out[15]: alp      AxesSubplot(0.125, 0.749828; 0.227941x0.130172)
alt      AxesSubplot(0.398529, 0.749828; 0.227941x0.130172)
ast      AxesSubplot(0.672059, 0.749828; 0.227941x0.130172)
er-cv     AxesSubplot(0.125, 0.593621; 0.227941x0.130172)
erytrocyt AxesSubplot(0.398529, 0.593621; 0.227941x0.130172)
etytr     AxesSubplot(0.672059, 0.593621; 0.227941x0.130172)
hbver     AxesSubplot(0.125, 0.437414; 0.227941x0.130172)
hematokrit AxesSubplot(0.398529, 0.437414; 0.227941x0.130172)
hemoglobin AxesSubplot(0.672059, 0.437414; 0.227941x0.130172)
indicator   AxesSubplot(0.125, 0.281207; 0.227941x0.130172)
leukocyty   AxesSubplot(0.398529, 0.281207; 0.227941x0.130172)
smoker      AxesSubplot(0.672059, 0.281207; 0.227941x0.130172)
trombocyty  AxesSubplot(0.125, 0.125; 0.227941x0.130172)
weight      AxesSubplot(0.398529, 0.125; 0.227941x0.130172)
dtype: object
```







## Hmotnosť

Podľa opisu numerických atribútov sme si všimli hneď jednu zvláštnosť. Minimálna hmotnosť je záporné číslo, čo v reálnom svete nie je možné. Preto je možné, že hmotnosť nie je uvedená vo všetkých záznamoch správne.

## Deskriptívna štatistika

In [16]:

```
labor.describe()
```

Out[16]:

	index	weight	hemoglobin	alp	etytr	alt	erytrocyt
<b>count</b>	10002.000000	10002.000000	9972.000000	9972.000000	9972.000000	9971.000000	9972.000000
<b>mean</b>	5000.500000	70.058861	7.605314	49.153497	7.199437	5.044637	7.617145
<b>std</b>	2887.47303	34.958888	1.776103	26.764248	1.820026	5.034543	1.014691
<b>min</b>	0.000000	-54.420260	1.094900	0.000000	0.000000	0.000000	3.748700
<b>25%</b>	2500.250000	46.000222	6.312152	22.855040	5.915890	2.388415	6.939980
<b>50%</b>	5000.500000	69.549650	7.696805	48.204640	7.213465	3.594600	7.625955
<b>75%</b>	7500.750000	93.765605	8.926235	75.731702	8.536662	5.812480	8.299573
<b>max</b>	10001.000000	196.504820	13.406460	100.000000	12.810620	100.000000	11.374090

Na základe deskriptívnych štatistik sa nám potvrdilo, že niektoré hodnoty sú naozaj nesprávne. Napríklad minimálna váha osoby je približne -54, čo nie je možné.

## Chýbajúce hodnoty

In [17]:

```
labor.shape[0] - labor.dropna().shape[0]
```

Out[17]:

330

In [18]:

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

Out[18]:

index	0
-------	---

```
weight          0
ssn            0
hemoglobin    30
alp            30
etytr          30
alt             31
erytroczyty   30
hbver          30
name            0
hematokrit    30
indicator       0
er-cv           30
leukocyty      30
smoker          0
relationship    0
ast              30
trombocyty     30
dtype: int64
```

```
In [19]: labor.isnull().sum().sum()
```

```
Out[19]: 331
```

```
In [20]: print("Chýbajúce dátovia {:.3f}% dát".format(labor.isnull().sum().sum() / labor.
```

Chýbajúce dátovia 3.309% dát

Nahradiť chýbajúce atribúty mediánmi alebo priemermi. Chýbajúce hodnoty predstavujú zhruba 3,3% všetkých hodnôt a preto ich nahradenie mediánmi pravdepodobne nespôsobí žiadne tăžkosti.

## DATASET Profiles.csv

Dataset Profiles obsahuje informácie o pacientoch z datasetu Labor.

```
In [21]: profiles.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3097 entries, 0 to 3096
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype  
 --- 
 0   Unnamed: 0   3097 non-null   int64  
 1   race         3097 non-null   object  
 2   residence    3097 non-null   object  
 3   job          3097 non-null   object  
 4   birthdate    3097 non-null   object  
 5   company      3097 non-null   object  
 6   blood_group  3097 non-null   object  
 7   sex          3097 non-null   object  
 8   ssn          3097 non-null   object  
 9   name         3097 non-null   object  
dtypes: int64(1), object(9)
memory usage: 242.1+ KB
```

Pomocou tohto výpisu sa dajú zistiť všetky potrebné informácie:

- Počet záznamov (2. riadok): 3097 (0 - 3096)
- Počet atribútov (3. riadok): 10 (0 - 9)

- Typy atribútov (4. stĺpec v tabuľke): int64, object

Okrem iného sa dá z toho vyčítať, že všetky zápisy nemajú žiadne NaN hodnoty, lebo počet všetkých záznamov sa rovná počtu hodnôt pre každý atribút.

## Duplikáty

```
In [22]: profiles.duplicated().any()
```

```
Out[22]: False
```

To znamená, že v dátach nie sú žiadne duplicitné záznamy.

```
In [23]: labor.nunique()
```

```
Out[23]: index      10002
          weight     9902
          ssn        3097
          hemoglobin 9793
          alp        9868
          etytr      9802
          alt        9805
          erytrocyty 9753
          hbver      9744
          name       3034
          hematokrit 9800
          indicator    2
          er-cv       9859
          leukocyty   9784
          smoker      2
          relationship 6
          ast         9867
          trombocyty  9740
          dtype: int64
```

```
In [24]: profiles.nunique()
```

```
Out[24]: Unnamed: 0      3097
          race        8
          residence  3097
          job        630
          birthdate  3058
          company    2914
          blood_group 8
          sex        2
          ssn        3097
          name       3034
          dtype: int64
```

Na základe toho, že počet ssn sa zhoduje s počtom riadkov v tabuľke profiles, môžeme usúdiť, že žiadne duplicitné záznamy sa v tabuľke profiles nenachádzajú.

## Kategórické atribúty

**race**

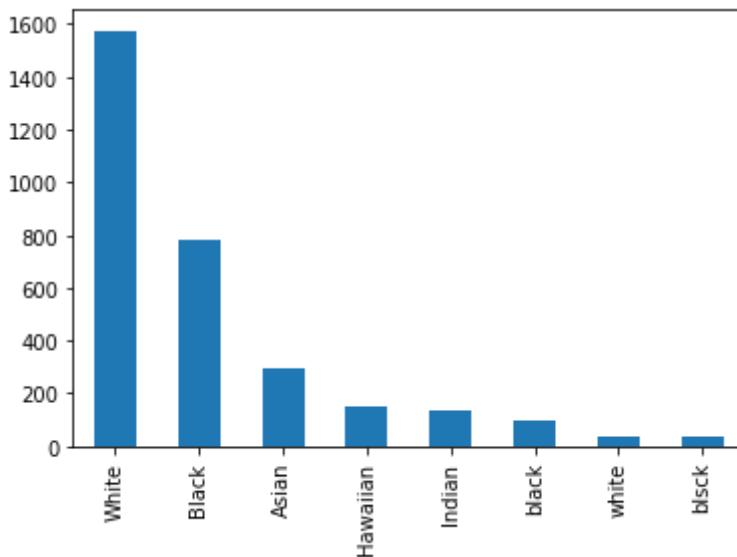
```
In [25]: profiles['race'].value_counts()
```

```
Out[25]: White      1576
          Black     780
```

```
Asian      296
Hawaiian   152
Indian     135
black      93
white      33
blsck      32
Name: race, dtype: int64
```

```
In [26]: profiles['race'].value_counts().plot(kind='bar')
```

```
Out[26]: <AxesSubplot:>
```



Tento atribút obsahuje nejednotné formáty: White - white, Black - black, blsck (zjavne ide iba o preklep pri písaní). Preto môžeme nejednotné formáty zlúčiť do jedného formátu.

```
In [27]: profiles["race"].replace({"black": "Black", "white": "White", "blsck": "Black"}, inplace=True)
```

```
In [28]: profiles['race'].value_counts()
```

```
Out[28]: White      1609
Black      905
Asian      296
Hawaiian   152
Indian     135
Name: race, dtype: int64
```

**birthday**

```
In [29]: profiles['birthdate'].value_counts()
```

```
Out[29]: 01/14/1908, 00:00:00      2
02/04/1968, 00:00:00      2
1927-08-28                  2
03/16/1970, 00:00:00      2
1970/10/19                  2
..
1983/04/16                  1
10 Dec 1956                  1
09 Dec 1981                  1
1946/11/15                  1
07/16/1922, 00:00:00      1
Name: birthdate, Length: 3058, dtype: int64
```

Už iba podľa malého náhľadu na hodnoty tohto atribútu vidno, že dátumy sú nejednotné.  
Vhodné by bolo ich mať v jednom formáte.

```
In [30]: profiles['birthdate'] = pd.to_datetime(profiles['birthdate'], utc=False)
```

```
In [31]: profiles['birthdate'].value_counts()
```

```
Out[31]: 1997-08-13    4  
2013-01-20    3  
1927-09-05    3  
1912-08-31    2  
1932-01-29    2  
..  
1945-04-22    1  
1984-08-06    1  
1950-07-18    1  
1947-10-13    1  
1922-07-16    1  
Name: birthdate, Length: 2956, dtype: int64
```

Takto upravené dátumy sú jednoduchšie na analýzu. Ešte skontrolujeme rozsah

```
In [32]: profiles['birthdate'].describe(datetime_is_numeric=True)
```

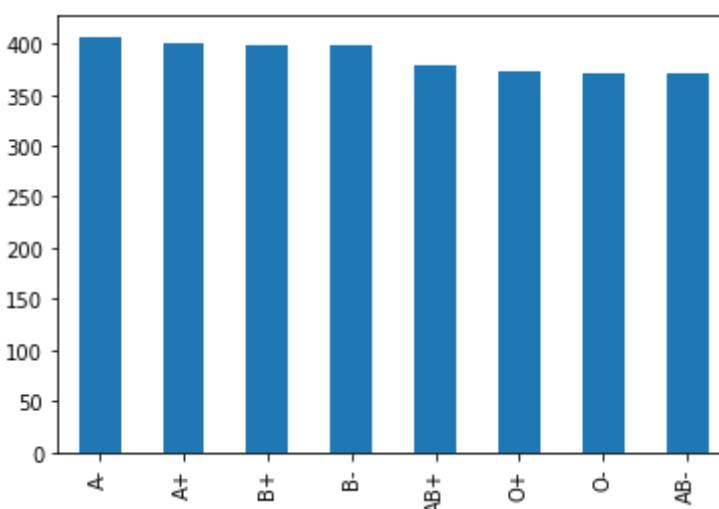
```
Out[32]: count                3097  
mean      1963-09-12 20:10:18.404907968  
min       1905-10-08 00:00:00  
25%       1933-08-07 00:00:00  
50%       1963-12-27 00:00:00  
75%       1992-09-23 00:00:00  
max       2021-09-14 00:00:00  
Name: birthdate, dtype: object
```

Rozsah zdá sa byť v norme, hoci je celkom na zváženie, či ľudia narodení v tomto roku môžu mať správny dátum narodenia.

### blood\_group

```
In [33]: profiles['blood_group'].value_counts().plot(kind='bar')
```

```
Out[33]: <AxesSubplot:
```



### sex

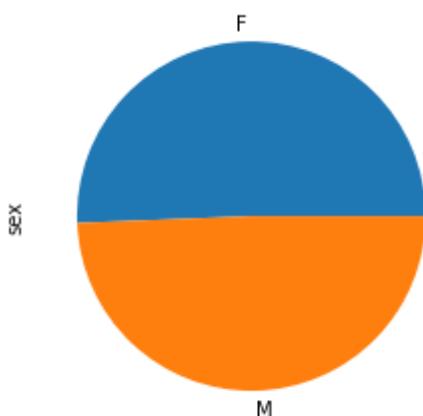
```
In [34]: profiles['sex'].value_counts()
```

```
Out[34]: F    1568  
M    1529  
Name: sex, dtype: int64
```

V tomto prípade, všetky typy atribútov majú jednotný formát. A je vidno, že rozdelenie pohlavia je približne rovnaké.

```
In [35]: profiles['sex'].value_counts().plot(kind='pie')
```

```
Out[35]: <AxesSubplot:ylabel='sex'>
```



### ssn

```
In [36]: profiles['ssn'].value_counts()
```

```
Out[36]: 087-49-2961    1  
049-29-3450    1  
413-78-6839    1  
490-12-7724    1  
668-35-1604    1  
..  
759-31-2559    1  
882-93-4577    1  
310-81-4482    1  
761-96-7670    1  
127-21-2115    1  
Name: ssn, Length: 3097, dtype: int64
```

Atribút SSN je Social Security Number, ktoré sa používa podobne ako u nás rodné číslo, čiže ide len o administratívny údaj.

## Numerické atribúty

Záznamy neobsahujú žiadne numerické atribúty okrem spomínaného prvého atribútu, ale ten možno pokladat za číslovanie.

```
In [37]: profiles = profiles.drop(profiles.columns[0], axis=1)
```

## Deskriptívna štatistika

```
In [38]: profiles.describe(datetime_is_numeric=True)
```

Out[38]:

	birthdate
count	3097
mean	1963-09-12 20:10:18.404907968
min	1905-10-08 00:00:00
25%	1933-08-07 00:00:00
50%	1963-12-27 00:00:00
75%	1992-09-23 00:00:00
max	2021-09-14 00:00:00

## Chýbajúce hodnoty

In [39]:

```
profiles.shape[0] - profiles.dropna().shape[0]
```

Out[39]:

```
0
```

In [40]:

```
profiles.isnull().sum()
```

Out[40]:

```
race          0
residence    0
job           0
birthdate     0
company       0
blood_group   0
sex           0
ssn           0
name          0
dtype: int64
```

In [41]:

```
profiles.isnull().sum().sum()
```

Out[41]:

```
0
```

Dataset profiles neobsahuje žiadne chýbajúce hodnoty.

## 2. Párová analýza dát (5 bodov)

- Preskúmajte vzťahy medzi zvolenými dvojicami atribútov.
- Identifikujte závislostí medzi dvojicami atribútov a závislosti medzi predikovanou premennou a ostatnými premennými.

## Skúmanie vzťahov medzi dvojicami atribútov

In [42]:

```
labor2 = labor.drop(["index", "name"], axis=1)
profiles2 = profiles.drop(["residence", "job", "company", "name"], axis=1)
```

In [43]:

```
merged = pd.merge(profiles2, labor2, how='outer', on='ssn')
```

In [44]:

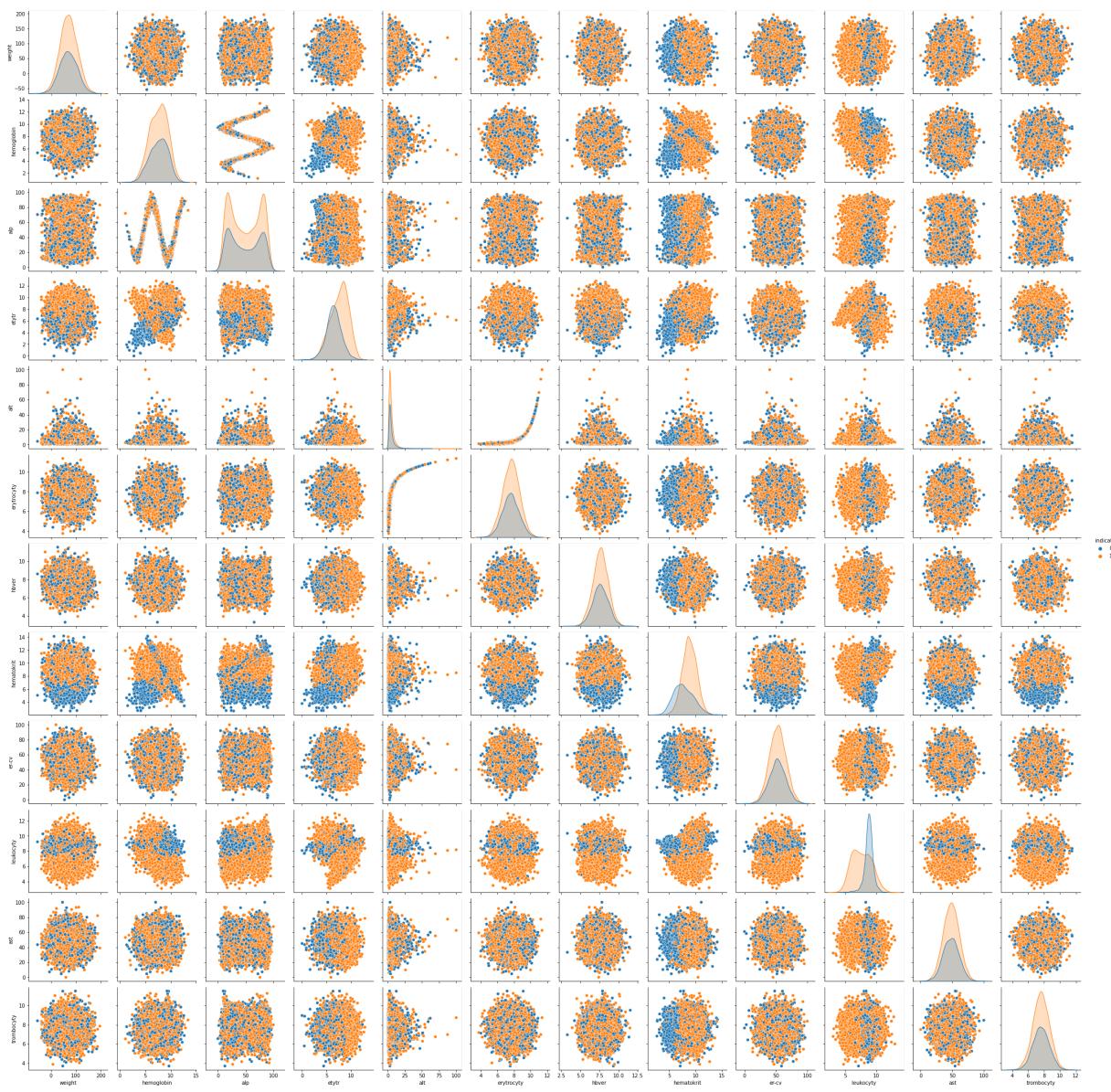
```
merged.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 10002 entries, 0 to 10001
Data columns (total 20 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   race        10002 non-null   object  
 1   birthdate    10002 non-null   datetime64[ns]
 2   blood_group  10002 non-null   object  
 3   sex          10002 non-null   object  
 4   ssn          10002 non-null   object  
 5   weight       10002 non-null   float64 
 6   hemoglobin   9972 non-null   float64 
 7   alp          9972 non-null   float64 
 8   etytr        9972 non-null   float64 
 9   alt          9971 non-null   float64 
 10  erytrocyty  9972 non-null   float64 
 11  hbver        9972 non-null   float64 
 12  hematokrit   9972 non-null   float64 
 13  indicator    10002 non-null   float64 
 14  er-cv         9972 non-null   float64 
 15  leukocyty    9972 non-null   float64 
 16  smoker        10002 non-null   int64  
 17  relationship  10002 non-null   object  
 18  ast          9972 non-null   float64 
 19  trombocyty   9972 non-null   float64 
dtypes: datetime64[ns](1), float64(13), int64(1), object(5)
memory usage: 1.6+ MB
```

Zlúčili sme oba datasety do jedného na základe SSN.

## Labor

```
In [45]: sns.pairplot(merged, hue="indicator", dropna=True, vars=['weight', 'hemoglobin', 'al
                           'erytrocyty', 'hbver', 'he
                           'er-cv', 'leukocyty', 'ast
Out[45]: <seaborn.axisgrid.PairGrid at 0x261110303a0>
```



Celkový pairplot nám umožňuje jednoducho vizuálne nájsť zaujímavé páry.

```
In [46]: labor.corr()
```

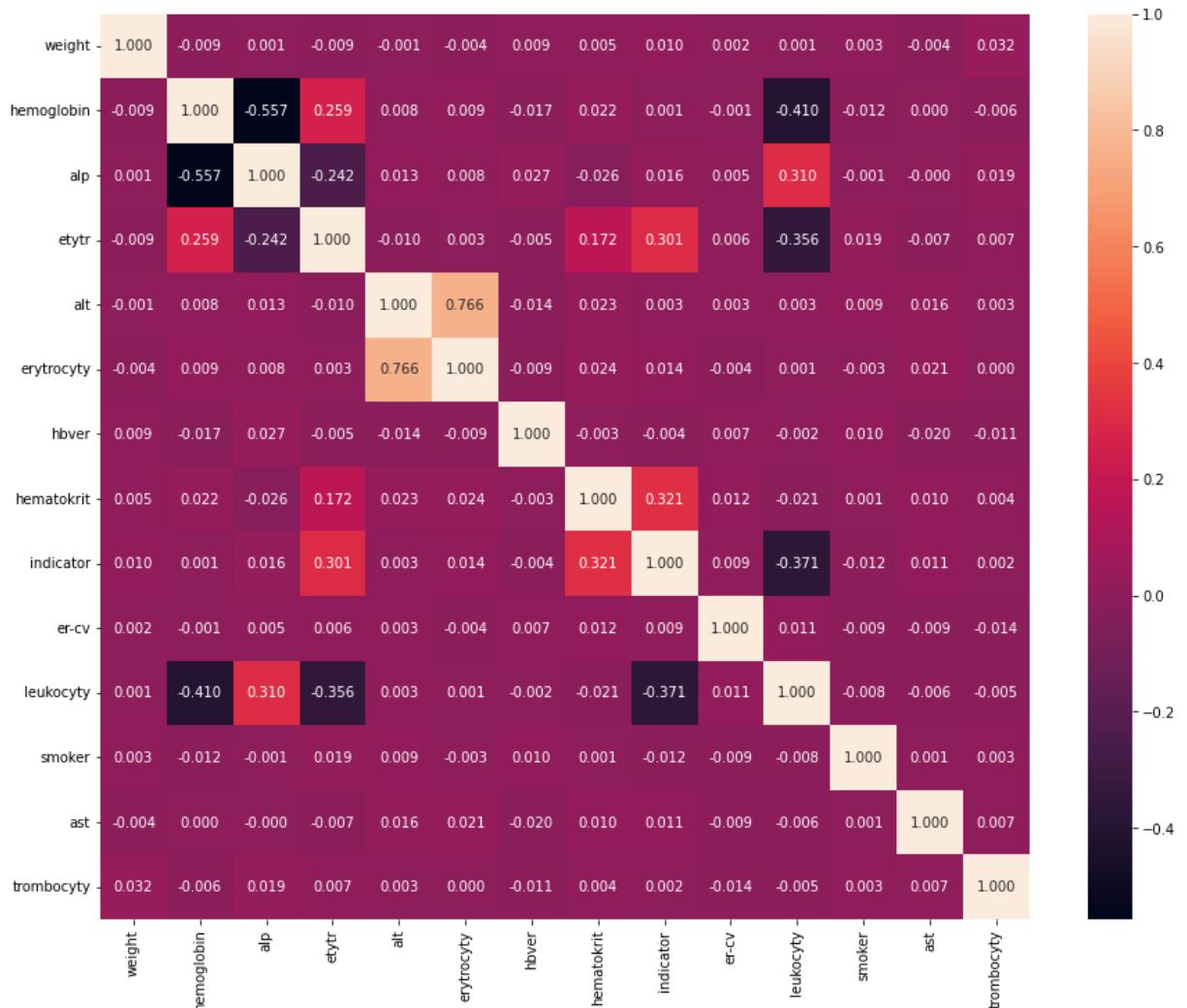
	<b>index</b>	<b>weight</b>	<b>hemoglobin</b>	<b>alp</b>	<b>etytr</b>	<b>alt</b>	<b>erytrocyt</b>	<b>hbve</b>
<b>index</b>	1.000000	0.012996	0.001189	-0.010044	0.021588	0.012657	0.018537	0.00406
<b>weight</b>	0.012996	1.000000	-0.008630	0.001016	-0.008526	-0.000766	-0.004177	0.00851
<b>hemoglobin</b>	0.001189	-0.008630	1.000000	-0.556709	0.259444	0.008399	0.008946	-0.01721
<b>alp</b>	-0.010044	0.001016	-0.556709	1.000000	-0.241561	0.012876	0.007804	0.02698
<b>etytr</b>	0.021588	-0.008526	0.259444	-0.241561	1.000000	-0.010173	0.002729	-0.00505
<b>alt</b>	0.012657	-0.000766	0.008399	0.012876	-0.010173	1.000000	0.766217	-0.01434
<b>erytrocyt</b>	0.018537	-0.004177	0.008946	0.007804	0.002729	0.766217	1.000000	-0.00903
<b>hbver</b>	0.004062	0.008512	-0.017214	0.026981	-0.005059	-0.014341	-0.009038	1.00000
<b>hematokrit</b>	-0.008957	0.005351	0.022012	-0.026477	0.171819	0.023043	0.024050	-0.00327
<b>indicator</b>	-0.008192	0.010240	0.001261	0.016469	0.300965	0.003407	0.013604	-0.00388
<b>er-cv</b>	-0.005742	0.002056	-0.000601	0.004906	0.005758	0.003356	-0.004037	0.00658

	<b>index</b>	<b>weight</b>	<b>hemoglobin</b>	<b>alp</b>	<b>etytr</b>	<b>alt</b>	<b>erytrocyt</b>	<b>hbve</b>
<b>leukocyty</b>	0.001026	0.000907	-0.409937	0.310392	-0.356013	0.002878	0.000578	-0.00236
<b>smoker</b>	-0.006369	0.003198	-0.012290	-0.001189	0.018910	0.009358	-0.002843	0.01010
<b>ast</b>	0.030712	-0.003890	0.000020	-0.000257	-0.007433	0.016120	0.021083	-0.01972
<b>trombocyty</b>	0.002249	0.031532	-0.006345	0.018788	0.007310	0.002622	0.000254	-0.01128

In [47]:

```
fig, ax = plt.subplots(figsize=(15,12))
sns.heatmap(merged.corr(), ax=ax, annot=True, fmt=".3f")
```

Out[47]: <AxesSubplot:>



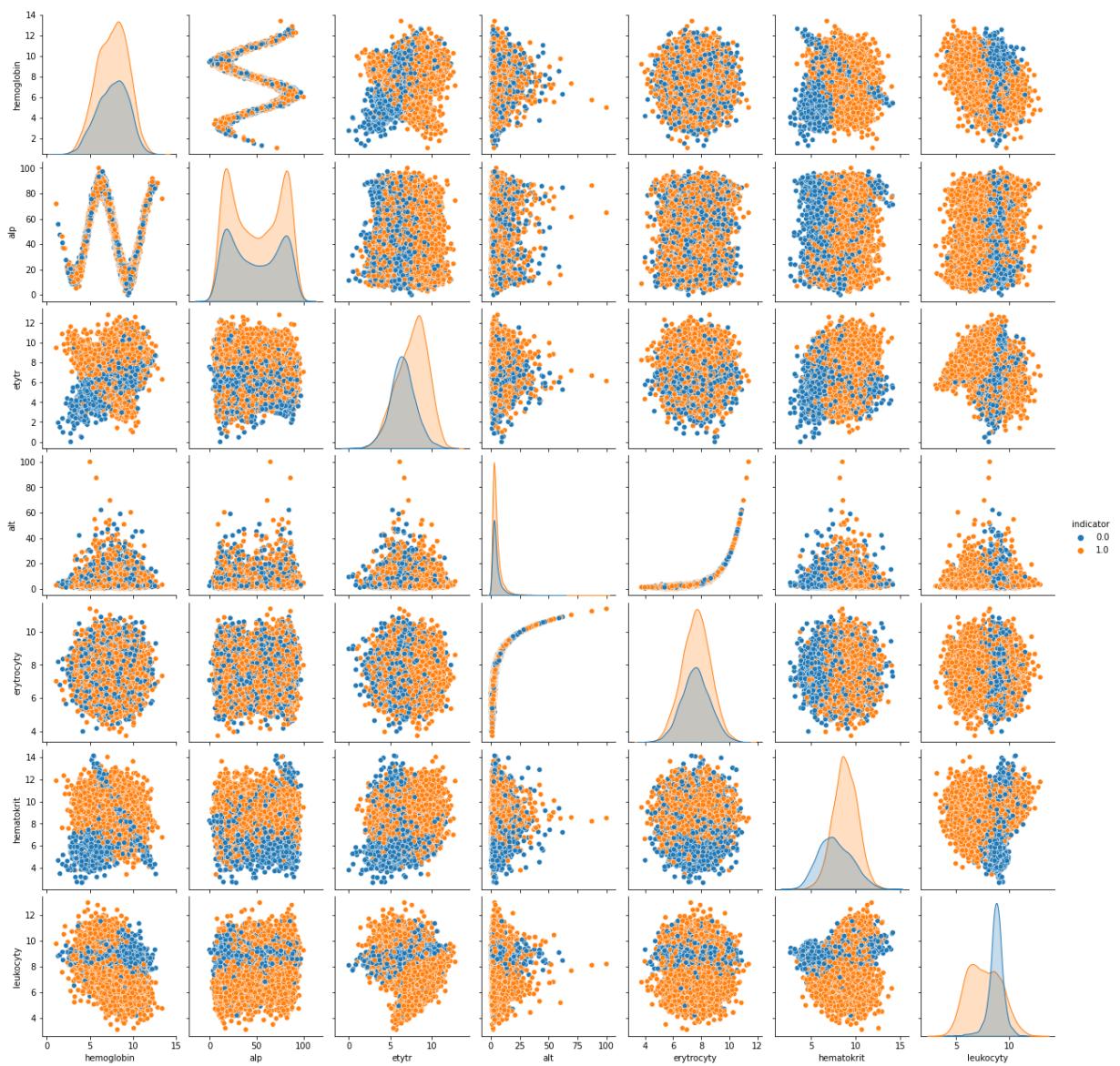
Na základe heatmapy korelácií vidíme, ktoré atribúty navzájom súvisia.

In [48]:

```
sns.pairplot(labor, hue="indicator", vars=['hemoglobin', 'alp', 'etytr', 'alt', 'ery
```

Out[48]:

<seaborn.axisgrid.PairGrid at 0x261239a6580>



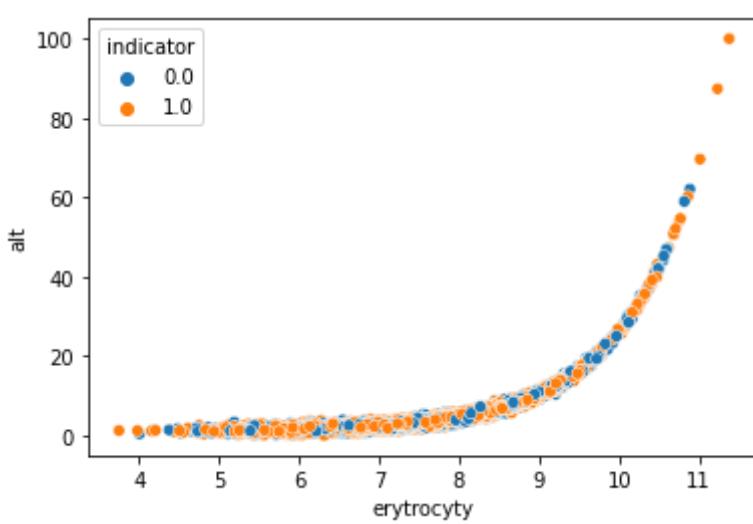
Tento pairplot obsahuje už iba atribúty ktorých korelácie môžu byť významné.

### Identifikácia najvýznamnejších závislostí medzi dvojicami atribútov na základe korelácií

In [49]:

```
sns.scatterplot(x=labor['erytrocyt'], y=labor['alt'], hue=labor['indicator'])
```

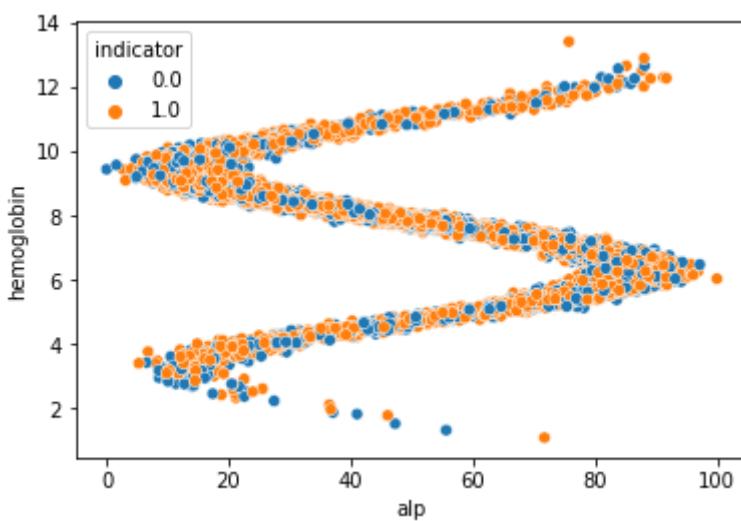
Out[49]:



Na tomto grafe vidno silnú kladnú koreláciu medzi erytrocytami a alt.

```
In [50]: sns.scatterplot(x=labor['alp'], y=labor['hemoglobin'], hue=labor['indicator'])
```

```
Out[50]: <AxesSubplot:xlabel='alp', ylabel='hemoglobin'>
```



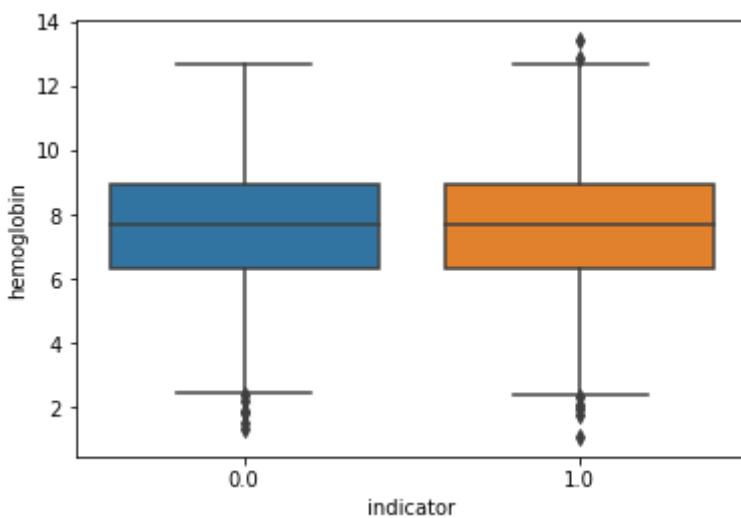
Hoci sa podľa výrazne zápornej hodnoty korelácie zdá byť tento pairplot dôležitý, v skutočnosti z neho nie je táto záporná korelácia vizuálne jasne viditeľná.

### Závislostí medzi predikovanou premennou a ostatnými premennými

Niektoré hodnoty nevykazujú významné rozdiely medzi pacientmi, ktorým sa stav zlepšil a medzi tými, ktorým sa nezlepšil. Tento takmer neexistujúci rozdiel demonštrujú dva príklady nižšie.

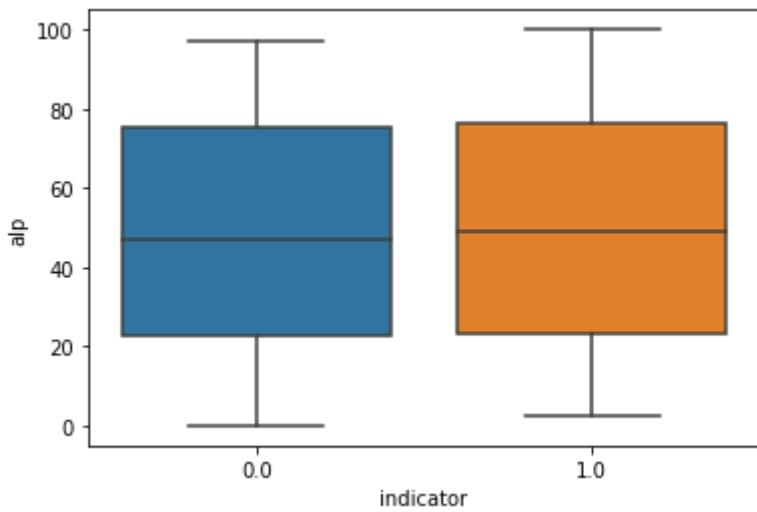
```
In [51]: sns.boxplot(x='indicator', y='hemoglobin', data=merged)
```

```
Out[51]: <AxesSubplot:xlabel='indicator', ylabel='hemoglobin'>
```



```
In [52]: sns.boxplot(x='indicator', y='alp', data=merged)
```

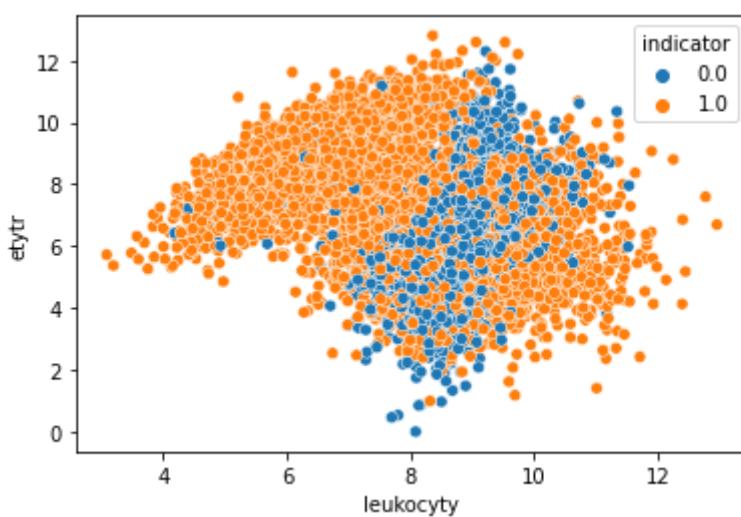
```
Out[52]: <AxesSubplot:xlabel='indicator', ylabel='alp'>
```



Niektoré hodnoty vykazujú rozdiely medzi pacientmi ktorým sa stav zlepšil a medzi tými, ktorým sa nezlepšil. Tie máme zobrazené na grafoch nižšie. O tom, či sú tieto rozdiely signifikantné, sformulujeme hypotézy.

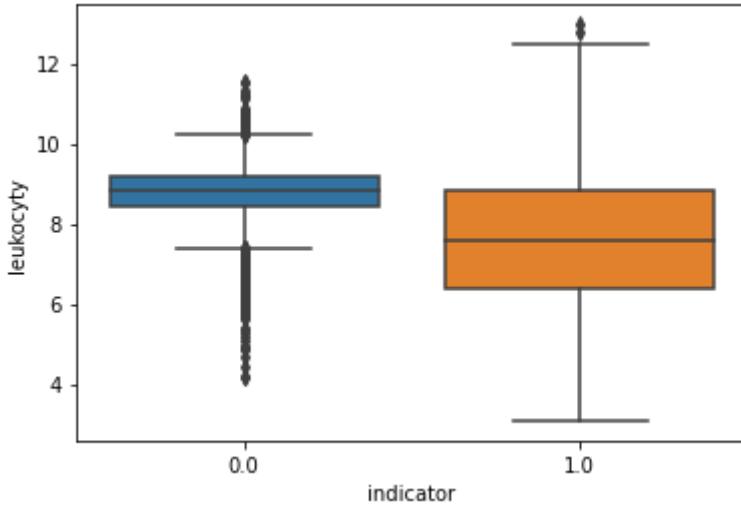
```
In [53]: sns.scatterplot(x=labor['leukocyty'], y=labor['etytr'], hue=labor['indicator'])
```

```
Out[53]: <AxesSubplot:xlabel='leukocyty', ylabel='etytr'>
```



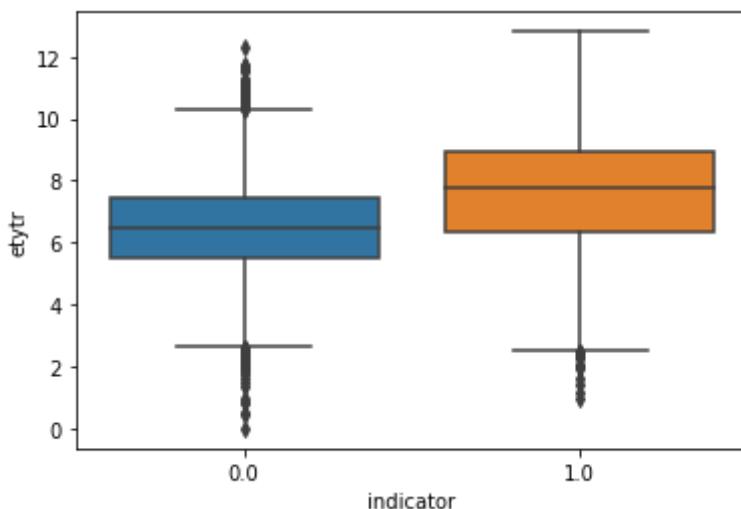
```
In [54]: sns.boxplot(x='indicator', y='leukocyty', data=merged)
```

```
Out[54]: <AxesSubplot:xlabel='indicator', ylabel='leukocyty'>
```



```
In [55]: sns.boxplot(x='indicator', y='etytr', data=merged)
```

```
Out[55]: <AxesSubplot:xlabel='indicator', ylabel='etytr'>
```



### 3. Formulácia a štatistické overenie hypotéz o dátach (2 body)

- Sformulujte dve hypotézy o dátach v kontexte zadanej predikčnej úlohy. Príkladom je napr. pacienti v zlepšenom stave majú v priemere inú (vyššiu/nižšiu) hodnotu nejakej látky alebo hormónu ako pacienti v nezlepšenom stave.
- Sformulované hypotézy overte vhodne zvoleným štatistickým testom.

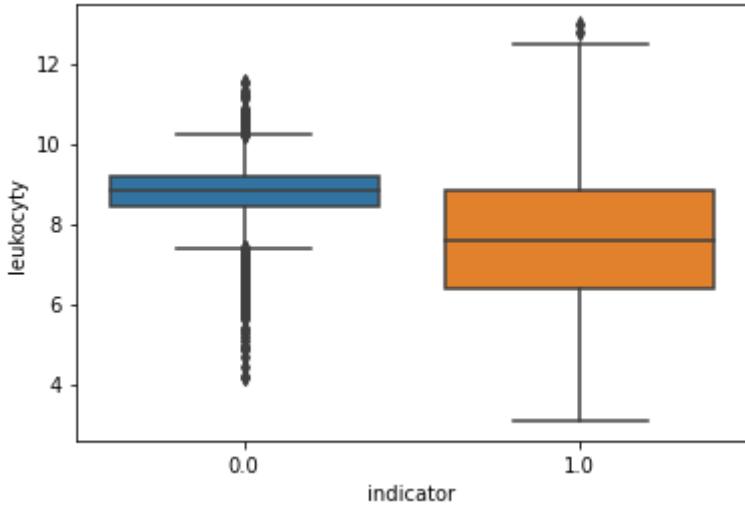
#### Hypotéza 1

$H_0$ : Hladina leukocytov u pacientov so zlepšeným stavom je v priemere rovnaká, ako hladina leukocytov u pacientov s nezlepšeným stavom.

$H_A$ : Rozdiel medzi hladinou leukocytov v krvi pri pacientoch so zlepšeným stavom oproti pacientom s nezlepšeným stavom je signifikantný.

```
In [56]: sns.boxplot(x='indicator', y='leukocyty', data=merged)
```

```
Out[56]: <AxesSubplot:xlabel='indicator', ylabel='leukocyty'>
```



```
In [57]: leukocyt_i = merged.loc[(merged['indicator'] == 1) & (merged['leukocyt'].notnull())]
```

```
In [58]: leukocyt_i.describe()
```

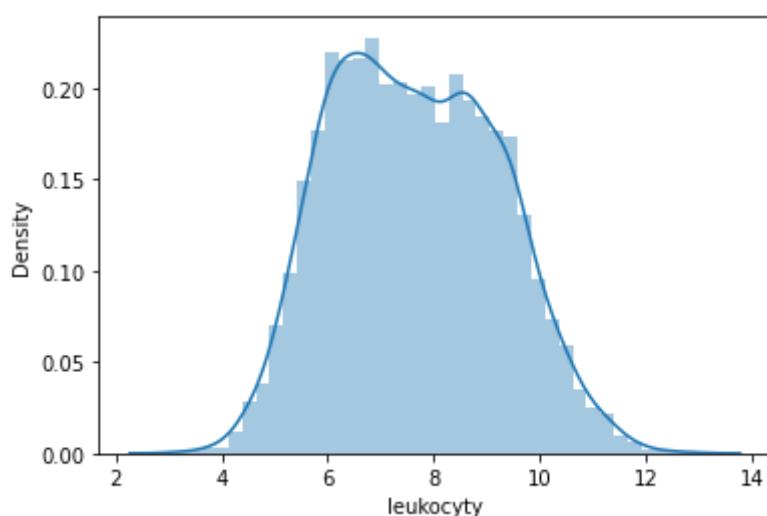
```
Out[58]: count    6403.000000
mean      7.649769
std       1.575718
min       3.083190
25%      6.395285
50%      7.576450
75%      8.839465
max      12.962710
Name: leukocyt, dtype: float64
```

```
In [59]: sns.distplot(leukocyt_i)
```

C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

```
Out[59]: <AxesSubplot:xlabel='leukocyt', ylabel='Density'>
```



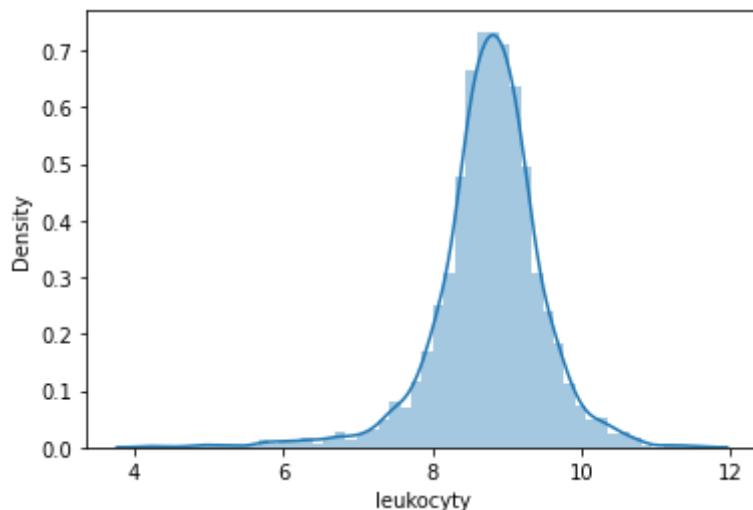
```
In [60]: leukocyt_ni = merged.loc[(merged['indicator'] == 0) & (merged['leukocyt'].notnull())]
```

```
In [61]: leukocyty_ni.describe()
```

```
Out[61]: count    3569.000000
mean      8.762914
std       0.722841
min       4.184570
25%      8.444770
50%      8.806140
75%      9.155660
max      11.539010
Name: leukocyty, dtype: float64
```

```
In [62]: sns.distplot(leukocyty_ni)
```

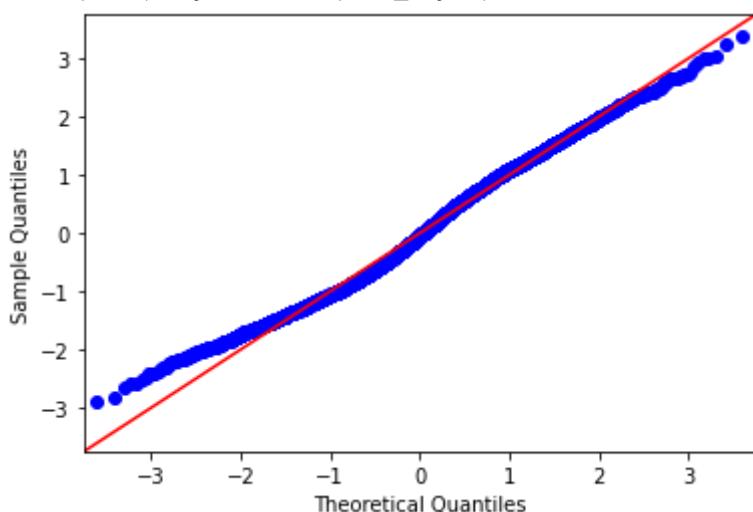
```
C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
<AxesSubplot:xlabel='leukocyty', ylabel='Density'>
```



```
In [63]: _ = sm.ProbPlot(leukocyty_i, fit=True).qqplot(line='45')
```

```
C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\statsmodels\graphics\gofplots.py:993: UserWarning: marker is redundantly defined by the 'marker' keyword argument and the fmt string "bo" (-> marker='o'). The keyword argument will take precedence.
```

```
ax.plot(x, y, fmt, **plot_style)
```

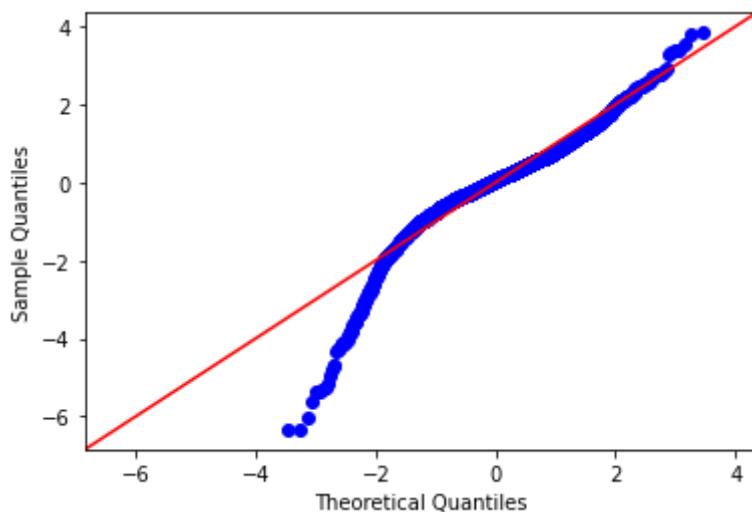


In [64]:

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

C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\statsmodels\graphics\gofplots.py:993: UserWarning: marker is redundantly defined by the 'marker' keyword argument and the fmt string "bo" (-> marker='o'). The keyword argument will take precedence.

```
ax.plot(x, y, fmt, **plot_style)
```



Podľa predchádzajúcich grafov nie je jednoznačne jasné, či pochádzajú obidva z normálneho rozdelenia, čo vychádza v nulovej hypotézy ( $H_0$ ). Preto použijeme Shapiro-Wilkov test normálnosti, aby sme si to overili.

In [65]:

```
stats.shapiro(leukocyty_i)
```

C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\scipy\stats\morestats.py:1760: UserWarning: p-value may not be accurate for N > 5000.

```
warnings.warn("p-value may not be accurate for N > 5000.")
```

Out[65]:

```
ShapiroResult(statistic=0.9890374541282654, pvalue=6.821046239747164e-22)
```

In [66]:

```
stats.shapiro(leukocyty_ni)
```

Out[66]:

```
ShapiroResult(statistic=0.9319184422492981, pvalue=1.3869885165110092e-37)
```

Na základe pvalue Shapiro-Wilkovho testu normálnosti môžeme prehlásiť, že dátá pochádzajú z iného, ako normálneho rozdelenia. 1. podmienka pre t-test nie je splnená. Varianciu netreba testovať, t-test sa použiť nedá. Keďže neboli splnené podmienky pre t-test, použijeme neparametrický Mann-Whitneyho U-test.

In [67]:

```
stats.levene(leukocyty_i, leukocyty_ni)
```

Out[67]:

```
LeveneResult(statistic=2638.812335978418, pvalue=0.0)
```

Na základe Levene testu vidíme, že pre t test nie je splnený ani predpoklad rovnakej variencie. Použijeme neparametrický Mann-Whitneyho U-test.

In [68]:

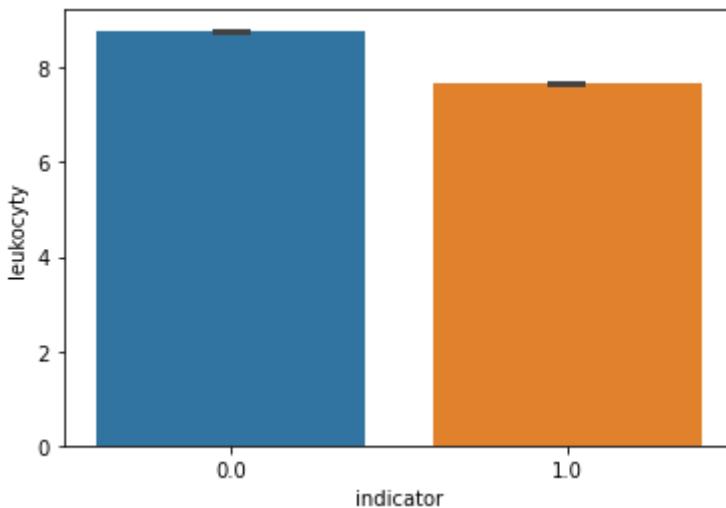
```
stats.mannwhitneyu(leukocyty_i, leukocyty_ni)
```

Out[68]:

```
MannwhitneyuResult(statistic=6260357.5, pvalue=1.6573645338571752e-307)
```

```
In [69]: sns.barplot(x='indicator', y='leukocyty', data=merged, capsize=0.1, errwidth=2)
```

```
Out[69]: <AxesSubplot:xlabel='indicator', ylabel='leukocyty'>
```

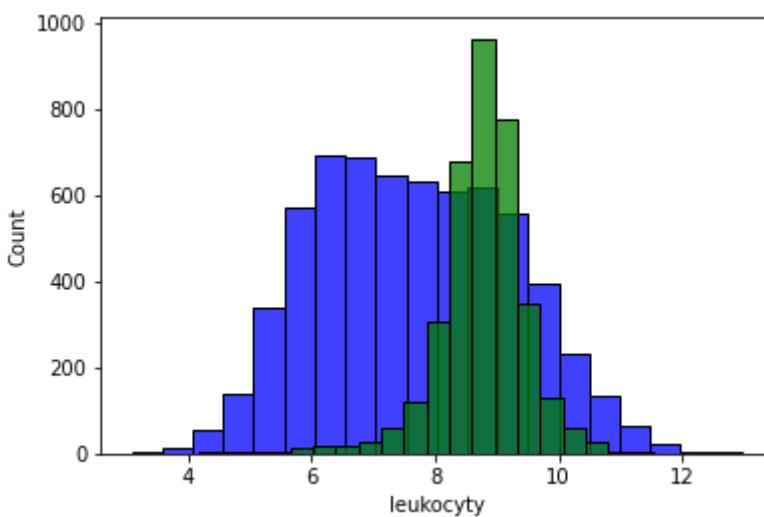


Podľa grafu vidno, že rozdiel medzi zlepšením a nezlepšením je zjavný.

A navyše tento graf povrdzuje, že odhad je dostatočne presný, ako vidno z errorbars.

```
In [70]: sns.histplot(leukocyty_i, bins=20, color='blue')  
sns.histplot(leukocyty_ni, bins=20, color='green')
```

```
Out[70]: <AxesSubplot:xlabel='leukocyty', ylabel='Count'>
```



Z hodnoty pvalue < 0.001 vyplýva, že pravdepodobnosť chyby 1. rádu je zanedbateľná (t.j. že je  $H_0$  pravdivá a my ju zamietneme). Z čoho môžeme skonštatovať, že  $H_0$  zamietame v prospech  $H_A$ .

Čo znamená, že rozdiel medzi hladinou leukocytov v krvi pri pacientoch so zlepšeným stavom oproti pacientom s nezlepšeným stavom **je signifikantný**.

## Sila testu

```
In [71]: def cohen_d(x1, x2):  
    nx1 = len(x1)  
    nx2 = len(x2)  
    s = np.sqrt(((nx1-1) * np.std(x1, ddof=1)**2 + (nx2-1) * np.std(x2, ddof=1)**2))  
    return (np.abs(np.mean(x1) - np.mean(x2))) / s
```

```
In [72]: leukocyty_c_d = cohen_d(leukocyty_i, leukocyty_ni)
```

```
Out[72]: 0.8340292150526264
```

```
In [73]: sm.stats.power.tt_ind_solve_power(leukocyty_c_d, len(leukocyty_i), 0.05, None, 1)
```

```
Out[73]: 1.0
```

Overili sme, že sila testu je dostatočná pre 1. hypotézu.

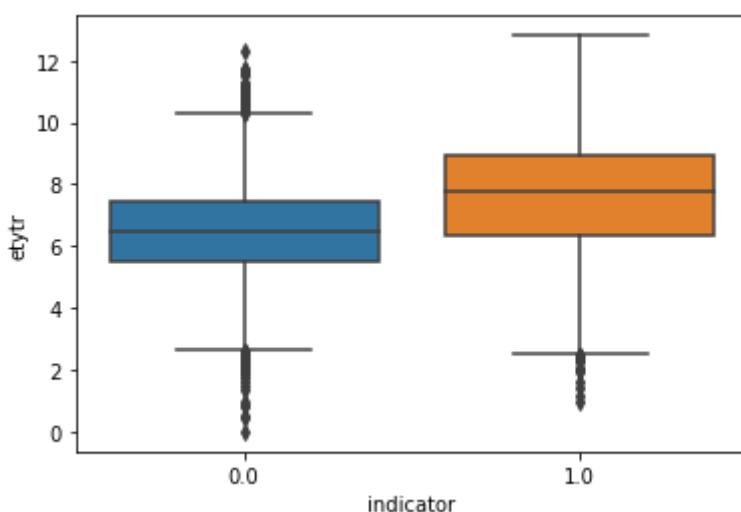
## Hypotéza 2

$H_0$ : Hladina etytr u pacientov so zlepšeným stavom je v priemere rovnaká, ako hladina leukocytov u pacientov s nezlepšeným stavom.

$H_A$ : Rozdiel medzi hladinou etytr v krvi pri pacientoch so zlepšeným stavom oproti pacientom s nezlepšeným stavom je signifikantný.

```
In [74]: sns.boxplot(x='indicator', y='etytr', data=merged)
```

```
Out[74]: <AxesSubplot:xlabel='indicator', ylabel='etytr'>
```



```
In [75]: etytr_i = merged.loc[(merged.indicator == 1) & (merged['etytr'].notnull()), 'etytr']
```

```
In [76]: etytr_i.describe()
```

```
Out[76]: count    6401.000000
mean      7.608550
std       1.826952
min       0.989410
25%      6.358910
50%      7.803530
75%      8.943100
max      12.810620
Name: etytr, dtype: float64
```

```
In [77]: etytr_ni = merged.loc[(merged.indicator == 0) & (merged['etytr'].notnull()), 'etytr']
```

```
In [78]: etytr_ni.describe()
```

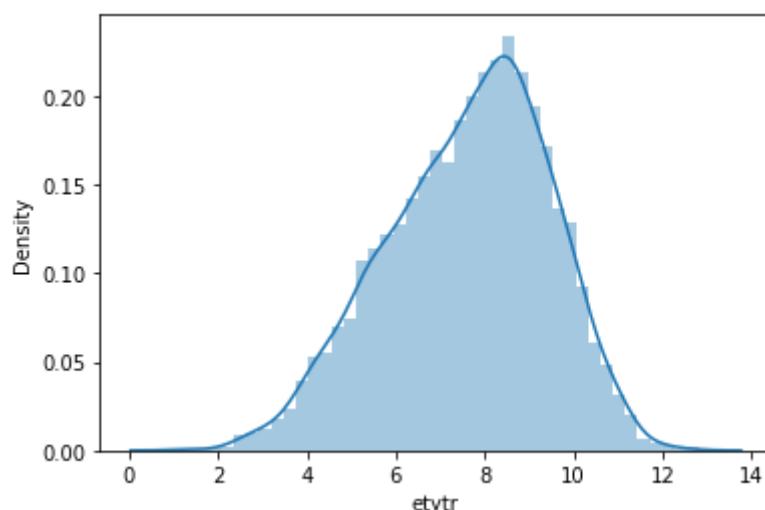
```
Out[78]: count    3571.000000
mean      6.466104
std       1.558882
min       0.000000
25%      5.504415
50%      6.446300
75%      7.423200
max     12.299010
Name: etytr, dtype: float64
```

```
In [79]: sns.distplot(etytr_i)
```

```
C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
```

```
    warnings.warn(msg, FutureWarning)
```

```
Out[79]: <AxesSubplot:xlabel='etytr', ylabel='Density'>
```

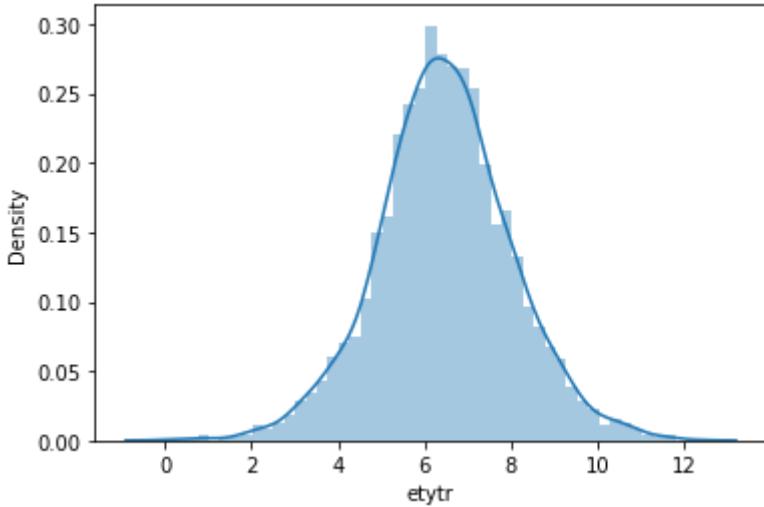


```
In [80]: sns.distplot(etytr_ni)
```

```
C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
```

```
    warnings.warn(msg, FutureWarning)
```

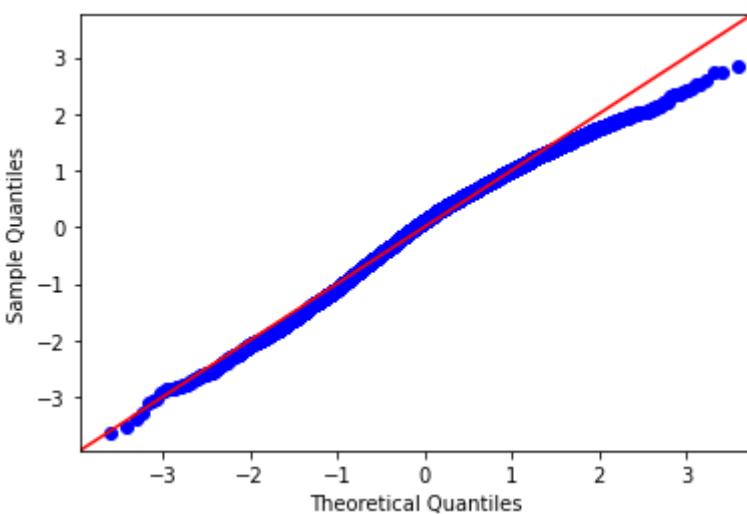
```
Out[80]: <AxesSubplot:xlabel='etyl', ylabel='Density'>
```



```
In [81]: _ = sm.ProbPlot(etytr_i, fit=True).qqplot(line='45')
```

C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\statsmodels\graphics\gofplots.py:993: UserWarning: marker is redundantly defined by the 'marker' keyword argument and the fmt string "bo" (-> marker='o'). The keyword argument will take precedence.

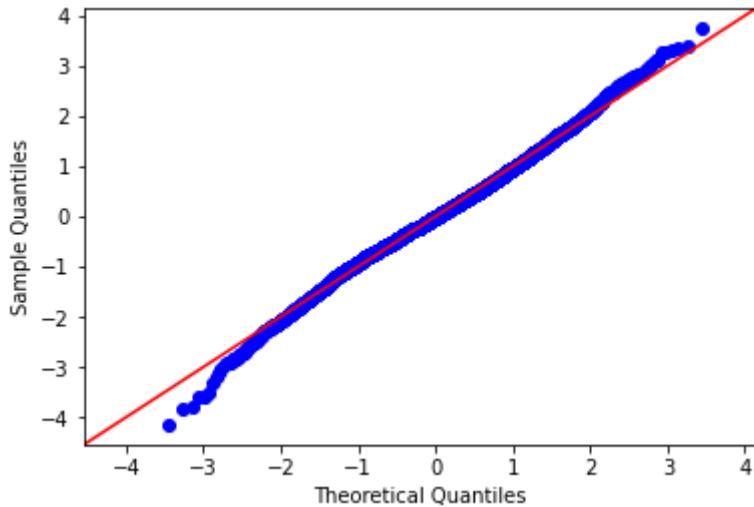
```
    ax.plot(x, y, fmt, **plot_style)
```



```
In [82]: _ = sm.ProbPlot(etytr_ni, fit=True).qqplot(line='45')
```

C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\statsmodels\graphics\gofplots.py:993: UserWarning: marker is redundantly defined by the 'marker' keyword argument and the fmt string "bo" (-> marker='o'). The keyword argument will take precedence.

```
    ax.plot(x, y, fmt, **plot_style)
```



Podľa predchádzajúcich grafov nie je jednoznačne jasné, či pochádzajú obidva z normálneho rozdelenia, čo vychádza v nulovej hypotézy ( $H_0$ ). Preto rovnako použijeme Shapiro-Wilkov test normálnosti, aby sme si to overili.

In [83]: `stats.shapiro(etytr_i)`

```
C:\Users\PeterSmrecek\Documents\IAU-repository\IAU-virtual\lib\site-packages\scipy\stats\morestats.py:1760: UserWarning: p-value may not be accurate for N > 5000.
    warnings.warn("p-value may not be accurate for N > 5000.")
Out[83]: ShapiroResult(statistic=0.9883873462677002, pvalue=1.5639635713839898e-22)
```

In [84]: `stats.shapiro(etytr_ni)`

```
Out[84]: ShapiroResult(statistic=0.9962351322174072, pvalue=7.766286103105813e-08)
```

Na základe pvalue Shapiro-Wilkovho testu normálnosti môžeme prehlásiť, že dátá pochádzajú z iného, ako normálneho rozdelenia. 1. podmienka pre t-test nie je splnená. Varianciu netreba testovať, t-test sa použiť nedá. Keďže neboli splnené podmienky pre t-test, použijeme neparametrický Mann-Whitneyho U-test.

In [85]: `stats.levene(etytr_i, etytr_ni)`

```
Out[85]: LeveneResult(statistic=158.5869244121984, pvalue=4.333365149085027e-36)
```

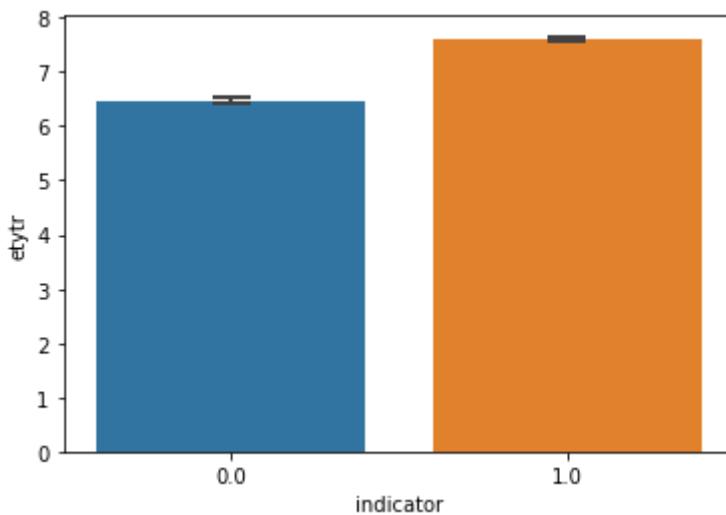
Na základe Levene testu vidíme, že pre t test nie je splnený ani predpoklad rovnakej variencie. Použijeme neparametrický Mann-Whitneyho U-test.

In [86]: `stats.mannwhitneyu(etytr_i, etytr_ni)`

```
Out[86]: MannwhitneyuResult(statistic=15772161.0, pvalue=6.083991905805457e-218)
```

In [87]: `sns.barplot(x='indicator', y='etytr', data=merged, capsizer=0.1, errwidth=2)`

```
Out[87]: <AxesSubplot:xlabel='indicator', ylabel='etytr'>
```

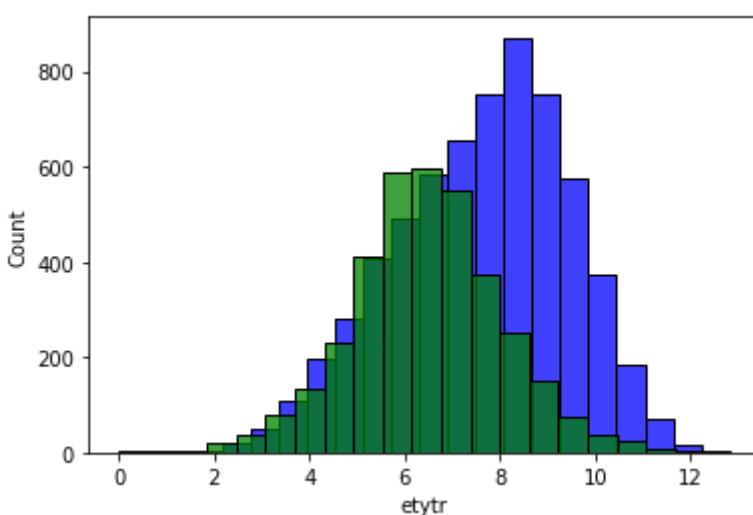


Podľa grafu vidno, podobne ako pri predchádzajúcej hypotéze, že rozdiel medzi zlepšením a nezlepšením je zjavný.

A tiež tento graf povrdzuje, že odhad je dostatočne presný, ako vidno z errorbars.

```
In [88]: sns.histplot(etytr_i, bins=20, color='blue')
sns.histplot(etytr_ni, bins=20, color='green')
```

```
Out[88]: <AxesSubplot:xlabel='etytr', ylabel='Count'>
```



Z hodnoty pvalue  $< 0.001$  vyplýva, že pravdepodobnosť chyby 1. rádu je zanedbateľná (t.j. že je  $H_0$  pravdivá a my ju zamietneme). Z čoho môžeme skonštatovať, že  $H_0$  zamietame v prospech  $H_A$ .

Čo znamená, že rozdiel medzi hladinou etytr v krvi pri pacientoch so zlepšeným stavom oproti pacientom s nezlepšeným stavom **je signifikantný**.

## Sila testu

```
In [89]: etytr_c_d = cohen_d(etytr_i, etytr_ni)
etytr_c_d
```

```
Out[89]: 0.6581940567369543
```

```
In [90]: sm.stats.power.tt_ind_solve_power(etytr_c_d, len(etytr_i), 0.05, None, 1)
```

---

Out[90]: 1.0

Overili sme, že sila testu je dostatočná pre 2. hypotézu.

## 4. Identifikácia problémov v dátach s navrhnutým riešením (3 body)

- Identifikujte problémy v dátach napr.: nevhodná štruktúra dát, duplicitné záznamy, nejednotné formáty, chýbajúce hodnoty, vychýlené hodnoty. V dátach sa môžu nachádzať aj iné, tu nevymenané problémy.
- Navrhnuté riešenie prvotne realizujte na dátach.

### Problémy v našich dátach:

- Odstránenie explicitného indexu
  - Implicitne ho pridáva pandas
- Opravenie jednoduchých chýb
  - Nahradenie textových stĺpcov číselnými hodnotami (napríklad v stĺpci smoker)
  - Opravenie preklepu v stĺci race
- Opravenie dátumov
  - Zjednotenie formátu dátumov narodenia
- Unikátnosť
  - Na základe toho, že počet unikátnych ssn v tabuľke labor sa zhoduje s počtom riadkov v tabuľke profiles, môžeme usúdiť, že žiadne duplicitné záznamy sa v tabuľke profiles nenachádzajú
- Chyby s hodnotami
  - Chýbajúce hodnoty sa nachádzajú v datasete labor, ale v datasete profiles sa nenachádzajú
  - Chýbajúce hodnoty sme v tejto časti projektu neriešili. V budúcich verziach projektu budú pravdepodobne nahradené priemerom, keďže predstavujú iba zhruba 3% všetkých hodnôt.
  - Vychýlené hodnoty (napríklad weight) sú prítomné vo viacerých stĺpcoch, ich riešením sa budeme zaoberať v budúcej fáze projektu.

### Riešenie problémov:

- Index sme odstránili v časti 1
- Jednoduché chyby sme odstránili v časti 1
- Dátumy sme opravili v časti 1
- Unikátnosť sme kontrolovali v časti 1
- Chyby s hodnotami budeme riešiť v nasledujúcej fáze projektu

## Správa sa odovzdáva v 6. týždni semestra

- Na cvičení, dvojica svojmu cvičiacemu odprezentuje vykonanú prieskumnú analýzu v Jupyter Notebooku.
- Správu elektronicky odovzdá jeden člen z dvojice do systému AIS do nedele 31.10.2021 23:59.

