

Project tutoré 2 - Murat Simsek et Yuquan Dai

R pour la question 1 et Python pour la question 2

```
# Importer les données nécessaires
dm1 <- read.csv2("dm1.csv", header = TRUE, sep = ",")
dm2 <- read.csv2("dm2.csv", header = TRUE, sep = ",")
ie1 <- read.csv2("ie1.csv", header = TRUE, sep = ",")
ie2 <- read.csv2("ie2.csv", header = TRUE, sep = ",")
lb1 <- read.csv2("lb1.csv", header = TRUE, sep = ",")
lb2 <- read.csv2("lb2.csv", header = TRUE, sep = ",")

# Jointure les données
alldm <- full_join(dm1, dm2)

## Joining, by = c("STUDYID", "DOMAIN", "USUBJID", "EPOCH", "VISIT", "VISITNUM",
## "RFSTDTC", "RFENDTC", "SITEID", "BRTHDTC", "AGE", "AGEU", "SEX", "RACE",
## "ETHNIC", "ARMCD", "ARM", "COUNTRY", "DMDTC", "DMDY")

allie <- full_join(ie1, ie2)

## Joining, by = c("STUDYID", "DOMAIN", "USUBJID", "EPOCH", "IESEQ", "IETESTCD",
## "IETEST", "IECAT", "IEORRES", "IESTRESC", "VISIT", "VISITNUM", "IEDTC", "IEDY")

alllb <- full_join(lb1, lb2)

## Joining, by = c("STUDYID", "DOMAIN", "USUBJID", "EPOCH", "LBSEQ", "LBTESTCD",
## "LBTEST", "LBCAT", "LBORRES", "LBORRESU", "LBORNRL0", "LBORNRI", "LBSTRESC",
## "LBSTRESN", "LBSTRESU", "LBSTNRLO", "LBSTNRHI", "LBNRIND", "LBSTAT",
## "LBREASND", "LBSPEC", "LBMETHOD", "LBBFL", "VISIT", "VISITNUM", "LBDTC",
## "LBDY")

# On a choisit dm pour travailler
n_distinct(alldm$USUBJID)

## [1] 411

# Faire une exclusion spéciale
allusrdm <- alldm$USUBJID[!(alldm$USUBJID %in% allie$USUBJID[allie$IECAT=="EXCLUSION"])]
summary(allusrdm)

##      Length      Class      Mode
##      369 character character
```

```
# Séparer BUPRENORPHINE/NALOXONE et CLONIDINE
allusrBN <- filter(alldm,
                  alldm$ARM=="BUPRENORPHINE/NALOXONE" &
                  alldm$USUBJID %in% allusrdm)
allusrC <- filter(alldm,
                  alldm$ARM=="CLONIDINE" &
                  alldm$USUBJID %in% allusrdm)
n_distinct(allusrBN$USUBJID)
```

```
## [1] 233
```

```
n_distinct(allusrC$USUBJID)
```

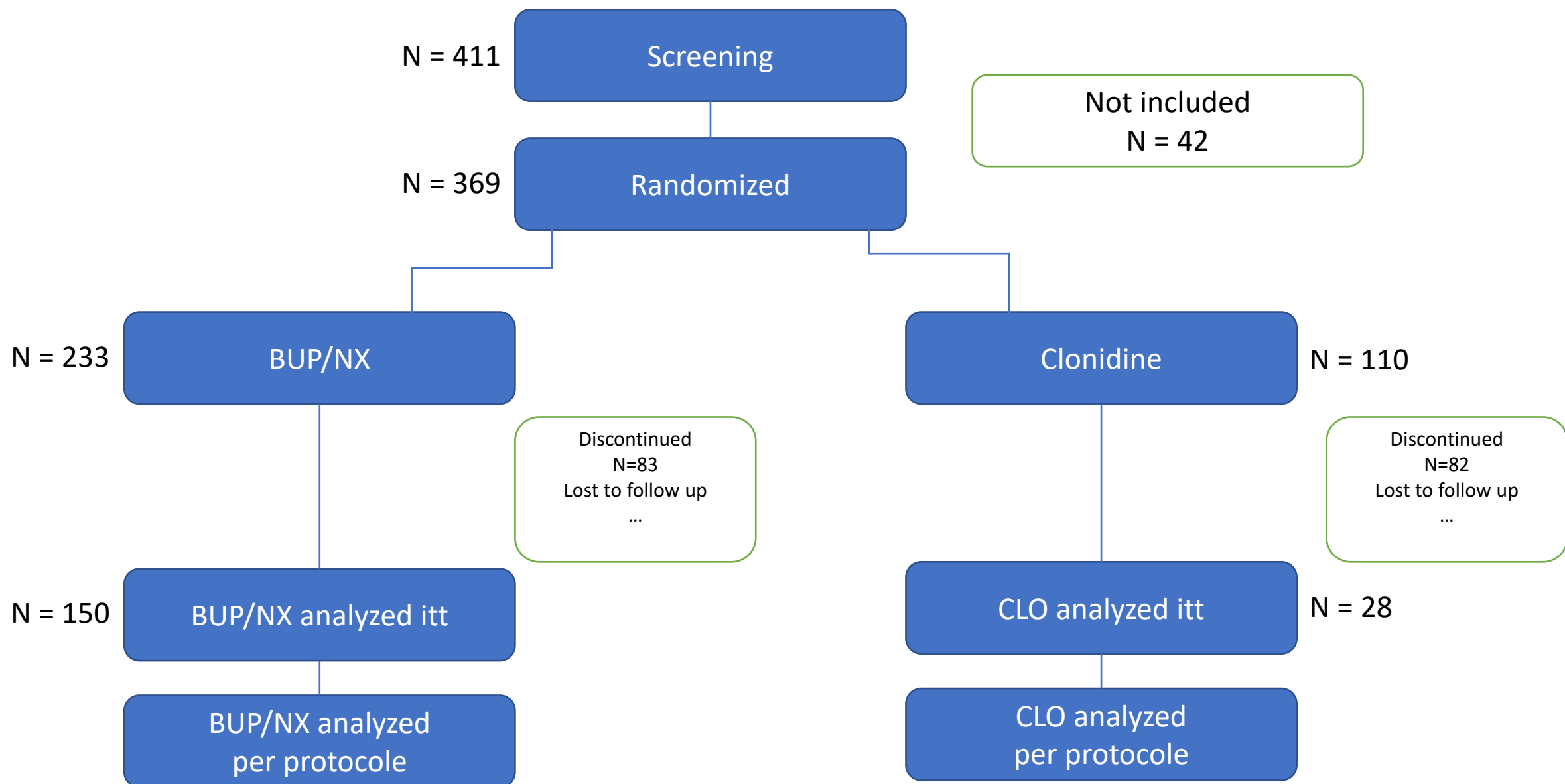
```
## [1] 110
```

```
# On considérera les patients qui ont des résultats au tests urinaires à J13 ou J14
usrBNfin <- filter(alllb,
                  (alllb$VISITNUM==13 | alllb$VISITNUM==14)&
                  (alllb$LBSTAT!="NOT DONE")&
                  (alllb$USUBJID %in% allusrBN$USUBJID))
usrCfin <- filter(alllb,
                  (alllb$VISITNUM==13 | alllb$VISITNUM==14)&
                  (alllb$LBSTAT!="NOT DONE")&
                  (alllb$USUBJID %in% allusrC$USUBJID))
n_distinct(usrBNfin$USUBJID)
```

```
## [1] 150
```

```
n_distinct(usrCfin$USUBJID)
```

```
## [1] 28
```



```
## PROJÉT TUTORÉ T-BEAU DES VARIABLES ##

In [115]:
import mitosheet
from mitosheet import *
import pandas as pd

In [116]:
# Subject Characteristics Dataset (SC)
df_sc_1 = pd.read_csv('C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_1/sc.csv')
df_sc_2 = pd.read_csv('C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_2/sc.csv')

df_sc = pd.concat([df_sc_1,df_sc_2])

# Pivoted df_sc
unused_columns = df_sc.columns.difference(set(['USUBJID']).union(set(['SCTEST'])).union(set(['SCORRES'])))
tmp_df = df_sc.drop(unused_columns, axis=1)
pivot_table = tmp_df.pivot_table(
    index=['USUBJID'],
    columns=['SCTEST'],
    values=['SCORRES'],
    aggfunc={'SCORRES': ['sum']}
)

# Flatten the column headers
pivot_table.columns = [make_valid_header(col) for col in pivot_table.columns.values]

# Reset the column name and the indexes
df_sc = pivot_table.rename_axis(None, axis=1).reset_index()

# Exposure Dataset (EX)
df_ex_1 = pd.read_csv('C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_1/ex.csv')
df_ex_2 = pd.read_csv('C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_2/ex.csv')

df_ex = pd.concat([df_ex_1,df_ex_2])

df_ex_ex=df_ex[['USUBJID','EXTRT']] # Selecting Extrt for Clodine and Burp
df_ex_ex.drop_duplicates(subset = 'USUBJID',keep = 'first', inplace = True)

# Demographics Dataset (DM)
df_dm_1 = pd.read_csv('C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_1/dm.csv')
df_dm_2 = pd.read_csv('C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_2/dm.csv')

df_dm = pd.concat([df_dm_1,df_dm_2])

df_dm = df_dm[['USUBJID','AGE','SEX','ARM']]
df_dm.drop_duplicates(subset = 'USUBJID',keep = 'first', inplace = True)
df_dm['AGE'] = df_dm['AGE'].str[:2] # age change

# Vital Signs Dataset (VS)
df_vs_1 = pd.read_csv('C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_1/vs.csv')
df_vs_2 = pd.read_csv('C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_2/vs.csv')

df_vs = pd.concat([df_vs_1,df_vs_2])

# Filtered VISIT in vs_csv
df_vs = df_vs[df_vs['VISIT'].str.contains('BASE', na=False)]
df_vs = df_vs.reset_index(drop=True)

# Pivoted vs_csv into df4
unused_columns = df_vs.columns.difference(set(['USUBJID']).union(set(['VSTEST'])).union(set(['VSORRES'])))
tmp_df = df_vs.drop(unused_columns, axis=1)
pivot_table = tmp_df.pivot_table(
    index=['USUBJID'],
    columns=['LBTEST'],
    values=['VSORRES'],
    aggfunc={'VSORRES': ['sum']}
)

# Flatten the column headers
pivot_table.columns = [make_valid_header(col) for col in pivot_table.columns.values]

# Reset the column name and the indexes
df_vs = pivot_table.rename_axis(None, axis=1).reset_index()

#Laboratory Tests Dataset (LB)
df_lb_1 = pd.read_csv(r'C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_1/lb.csv')
df_lb_2 = pd.read_csv(r'C:/Users/Casper/Desktop/Master1-Cours/Projet tutoré/ascii_2/lb.csv')

df_lb = pd.concat([df_lb_1,df_lb_2])

# Filtered VISIT in lb_csv
df_lb = df_lb[df_lb['VISIT'].str.contains('BASELINE', na=False)]
df_lb = df_lb.reset_index(drop=True)

# Pivoted lb_csv into df7
unused_columns = df_lb.columns.difference(set(['USUBJID']).union(set(['LBTEST'])).union(set(['LBORRES'])))
tmp_df = df_lb.drop(unused_columns, axis=1)
pivot_table = tmp_df.pivot_table(
    index=['USUBJID'],
    columns=['LBTEST'],
    values=['LBORRES'],
    aggfunc={'LBORRES': ['sum']}
)

# Flatten the column headers
pivot_table.columns = [make_valid_header(col) for col in pivot_table.columns.values]

# Reset the column name and the indexes
df_lb = pivot_table.rename_axis(None, axis=1).reset_index()

In [53]:
df_lb.columns

Out[53]:
Index(['USUBJID', 'LBORRES_sum_AMPHETAMINES', 'LBORRES_sum_BARBITURATES',
      'LBORRES_sum_BENZODIAZEPINES', 'LBORRES_sum_COCAINE',
      'LBORRES_sum_CREATININE', 'LBORRES_sum_METHADONE',
      'LBORRES_sum_METHAMPHETAMINE', 'LBORRES_sum_MORPHINE',
      'LBORRES_sum_NONE', 'LBORRES_sum_OPIATE', 'LBORRES_sum_PCP',
      'LBORRES_sum_TCA', 'LBORRES_sum_THC'],
      dtype='object')

In [54]:
df_lb.drop(['LBORRES_sum_AMPHETAMINES', 'LBORRES_sum_BARBITURATES',
            'LBORRES_sum_BENZODIAZEPINES', 'LBORRES_sum_COCAINE',
            'LBORRES_sum_CREATININE',
            'LBORRES_sum_METHAMPHETAMINE',
            'LBORRES_sum_NONE', 'LBORRES_sum_PCP',
            'LBORRES_sum_TCA', 'LBORRES_sum_THC'], axis = 1,inplace=True)

In [55]:
# Merge Data
df = df_sc.merge(df_ex,on= 'USUBJID', how= 'right')
df = df.merge(df_dm,on= 'USUBJID')
df = df.merge(df_vs,on= 'USUBJID',how= 'left')
df = df.merge(df_lb,on= 'USUBJID',how= 'left')

Out[55]:
   USUBJID  SCORRES_sum_EDUCATION_COMPLETED  SCORRES_sum_MARITAL_STATUS  SCORRES_sum_MET_ALL_INCLUSION_NO_EXCLUSION_CRIT  343 rows x 20 columns

In [56]:
df.columns = ['ID', 'EDUCATION_COMPLETED',
              'MARITAL_STATUS',
              'MET_ALL_INCLUSION_NO_EXCLUSION_CRIT_',
              'EMPLOYMENT_PATTERN_PAST_3_YEARS',
              'EMPLOYMENT_PATTERN_PAST_30_DAYS', 'EXTRT', 'AGE',
              'SEX', 'ARM', 'DIASTOLIC_BLOOD_PRESSURE',
              'HEIGHT', 'PULSE', 'RESPIRATIONS',
              'SYSTOLIC_BLOOD_PRESSURE', 'TEMPERATURE',
              'WEIGHT', 'METHADONE', 'MORPHINE',
              'OPIATE']

In [57]:
df

Out[57]:
   ID  EDUCATION_COMPLETED  MARITAL_STATUS  MET_ALL_INCLUSION_NO_EXCLUSION_CRIT_  EMPLOYMENT_PATTERN_PAST_3_YEARS
0  01_000579              14          DIVORCED                                YES          FULL TIME (35+ HRS/WK)
1  01_001362              13          NEVER MARRIED                            YES          UNEMPLOYED
2  01_001490              14          NEVER MARRIED                            YES          PART TIME (IRREGULAR DAYWORK)
3  01_002199              13          NEVER MARRIED                            YES          PART TIME (REGULAR HOURS)
4  01_002844              11          NEVER MARRIED                            YES          STUDENT
...  ...
338 02_098074              12          DIVORCED                                YES          UNEMPLOYED
339 02_098425              12          DIVORCED                                YES          FULL TIME (35+ HRS/WK)
340 02_099053              11          NEVER MARRIED                            YES          PART TIME (IRREGULAR DAYWORK)
341 02_099368              11          NEVER MARRIED                            YES          PART TIME (IRREGULAR DAYWORK)
342 02_099926              13          NEVER MARRIED                            YES          FULL TIME (35+ HRS/WK)

343 rows x 20 columns

In [59]:
df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 343 entries, 0 to 342
Data columns (total 20 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   ID                                         343 non-null    object
1   EDUCATION_COMPLETED                       343 non-null    object
2   MARITAL_STATUS                           341 non-null    object
3   MET_ALL_INCLUSION_NO_EXCLUSION_CRIT_     341 non-null    object
4   EMPLOYMENT_PATTERN_PAST_3_YEARS           342 non-null    object
5   EMPLOYMENT_PATTERN_PAST_30_DAYS           343 non-null    object
6   EXTRT                                     343 non-null    object
7   AGE                                        343 non-null    object
8   SEX                                        343 non-null    object
9   ARM                                        343 non-null    object
10  DIASTOLIC_BLOOD_PRESSURE                  340 non-null    float64
11  HEIGHT                                    337 non-null    float64
12  PULSE                                     342 non-null    float64
13  RESPIRATIONS                             335 non-null    float64
14  SYSTOLIC_BLOOD_PRESSURE                   340 non-null    float64
15  TEMPERATURE                             335 non-null    float64
16  WEIGHT                                    338 non-null    float64
17  METHADONE                                343 non-null    object
18  MORPHINE                                 342 non-null    object
19  OPIATE                                   286 non-null    object
dtypes: float64(7), object(13)
memory usage: 56.3+ KB

In [60]:
import matplotlib.pyplot as plt
import seaborn as sns
ax = sns.countplot(data = df, x = 'ARM')

ax.set_xticklabels(ax.get_xticklabels(), rotation=40, ha="right")
ax.bar_label(ax.containers[0])
sns.set(rc={'figure.figsize':(12,9)})
plt.tight_layout()
plt.show()

count
200
150
100
50
0
CLONIDINE
BUPRENORPHINE/NALOXONE
ARM
SCREEN FAILURE
233
107
3

In [61]:
df.columns

Out[61]:
Index(['ID', 'EDUCATION_COMPLETED', 'MARITAL_STATUS',
      'MET_ALL_INCLUSION_NO_EXCLUSION_CRIT_',
      'EMPLOYMENT_PATTERN_PAST_3_YEARS', 'EMPLOYMENT_PATTERN_PAST_30_DAYS',
      'EXTRT', 'AGE', 'SEX', 'ARM', 'DIASTOLIC_BLOOD_PRESSURE', 'HEIGHT',
      'PULSE', 'RESPIRATIONS', 'SYSTOLIC_BLOOD_PRESSURE', 'TEMPERATURE',
      'WEIGHT', 'METHADONE', 'MORPHINE', 'OPIATE'],
      dtype='object')

In [90]:
df['AGE'].replace(' ',NaN,inplace=True) # adjust age to be float and change ' ' to na

In [92]:
df['AGE'] = df['AGE'].astype(float)

In [97]:
# Change Weight and Height to BMI
# Formula BMI = weight (lb) / (height (in))2 x 703
df2 = df[df['HEIGHT'].notna()]
df2 = df[df['WEIGHT'].notna()]
df2.drop(308,inplace = True)

df['BMI'] = df2['WEIGHT'].astype(int) / (df2['HEIGHT'].astype(int)**2) * 703

C:/Users/Casper/anaconda3/lib/site-packages/pandas/core/frame.py:4906: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

In [98]:
df

Out[98]:
   ID  EDUCATION_COMPLETED  MARITAL_STATUS  MET_ALL_INCLUSION_NO_EXCLUSION_CRIT_  EMPLOYMENT_PATTERN_PAST_3_YEARS
0  01_000579              14          DIVORCED                                YES          FULL TIME (35+ HRS/WK)
1  01_001362              13          NEVER MARRIED                            YES          UNEMPLOYED
2  01_001490              14          NEVER MARRIED                            YES          PART TIME (IRREGULAR DAYWORK)
3  01_002199              13          NEVER MARRIED                            YES          PART TIME (REGULAR HOURS)
4  01_002844              11          NEVER MARRIED                            YES          STUDENT
...  ...
338 02_098074              12          DIVORCED                                YES          UNEMPLOYED
339 02_098425              12          DIVORCED                                YES          FULL TIME (35+ HRS/WK)
340 02_099053              11          NEVER MARRIED                            YES          PART TIME (IRREGULAR DAYWORK)
341 02_099368              11          NEVER MARRIED                            YES          PART TIME (IRREGULAR DAYWORK)
342 02_099926              13          NEVER MARRIED                            YES          FULL TIME (35+ HRS/WK)

343 rows x 21 columns

In [99]:
df.describe()

Out[99]:
   AGE  DIASTOLIC_BLOOD_PRESSURE  HEIGHT  PULSE  RESPIRATIONS  SYSTOLIC_BLOOD_PRESSURE  TEMPERATURE  WEIGH
count  340.000000                340.000000  337.000000  342.000000  335.000000          340.000000  335.000000  338.00000
mean   37.438235                83.720588   70.041543  155.254386  17.131343          126.485294  103.068955  166.66568
std    10.149284                40.081571   35.497963  32.364079  3.619732          32.514346   51.138584  35.71953
min    18.000000                50.000000   53.000000   64.000000  1.000000          94.000000   95.000000  100.00000
25%    29.000000                70.000000   66.000000  136.000000  16.000000          110.000000   98.000000  140.00000
50%    39.000000                79.000000   68.000000  152.000000  16.000000          120.000000   98.400000  162.00000
75%    45.000000                88.000000   71.000000  169.500000  18.000000          132.000000   98.900000  188.00000
max    65.000000                701.000000  715.000000  337.000000  36.000000          430.000000  999.900000  299.00000

In [102]:
from tableone import TableOne, load_dataset

In [103]:
df.columns

Out[103]:
Index(['ID', 'EDUCATION_COMPLETED', 'MARITAL_STATUS',
      'MET_ALL_INCLUSION_NO_EXCLUSION_CRIT_',
      'EMPLOYMENT_PATTERN_PAST_3_YEARS', 'EMPLOYMENT_PATTERN_PAST_30_DAYS',
      'EXTRT', 'AGE', 'SEX', 'ARM', 'DIASTOLIC_BLOOD_PRESSURE', 'HEIGHT',
      'PULSE', 'RESPIRATIONS', 'SYSTOLIC_BLOOD_PRESSURE', 'TEMPERATURE',
      'WEIGHT', 'METHADONE', 'MORPHINE', 'OPIATE', 'BMI'],
      dtype='object')

In [114]:
columns = ['EDUCATION_COMPLETED', 'MARITAL_STATUS',
            'EMPLOYMENT_PATTERN_PAST_3_YEARS', 'EMPLOYMENT_PATTERN_PAST_30_DAYS','AGE', 'ARM', 'DIASTOLIC_BLOOD_PRESSURE',
            'PULSE', 'RESPIRATIONS', 'SYSTOLIC_BLOOD_PRESSURE', 'TEMPERATURE','BMI', 'METHADONE', 'MORPHINE', 'OPIATE']

categorical = ['EDUCATION_COMPLETED', 'MARITAL_STATUS','EMPLOYMENT_PATTERN_PAST_3_YEARS', 'EMPLOYMENT_PATTERN_PAST_30_DAYS']

groupby = 'ARM'

mytable = TableOne(df, columns=columns, categorical = categorical,
                   groupby=groupby)
mytable

Out[114]:
Grouped by ARM

Missing Overall BUPRENORPHINE/NALOXONE CLONIDINE SCREEN FAILURE
n 343 233 107 3
EDUCATION_COMPLETED, n (%)
10 0 21 (6.1) 14 (6.0) 6 (5.6) 1 (33.3)
11 37 (10.8) 28 (12.0) 9 (8.4)
12 132 (38.5) 91 (39.1) 39 (36.4) 2 (66.7)
13 34 (9.9) 27 (11.6) 7 (6.5)
14 50 (14.6) 28 (12.0) 22 (20.6)
15 19 (5.5) 16 (6.9) 3 (2.8)
16 21 (6.1) 13 (5.6) 8 (7.5)
17 3 (0.9) 3 (1.3)
18 5 (1.5) 2 (0.9) 3 (2.8)
19 1 (0.3) 1 (0.4)
20 3 (0.9) 1 (0.4) 2 (1.9)
7 2 (0.6) 2 (0.9)
8 6 (1.7) 3 (1.3) 3 (2.8)
9 9 (2.6) 4 (1.7) 5 (4.7)
MARITAL_STATUS, n (%)
DIVORCED 0 55 (16.0) 38 (16.3) 17 (15.9)
LEGALLY MARRIED 61 (17.8) 47 (20.2) 14 (13.1)
LIVING WITH PARTNER/COHABITATING 33 (9.6) 20 (8.6) 12 (11.2) 1 (33.3)
NEVER MARRIED 160 (46.6) 109 (46.8) 49 (45.8) 2 (66.7)
SEPARATED 26 (7.6) 14 (6.0) 12 (11.2)
WIDOWED 8 (2.3) 5 (2.1) 3 (2.8)
EMPLOYMENT_PATTERN_PAST_30_DAYS, n (%)
FULL TIME (35+ HRS/WK) 1 189 (55.3) 127 (54.5) 61 (57.5) 1 (33.3)
HOMEMAKER 9 (2.6) 5 (2.1) 4 (3.8)
IN CONTROLLED ENVIRONMENT 2 (0.6) 2 (0.9)
PART TIME (IRREGULAR DAYWORK) 47 (13.7) 33 (14.2) 14 (13.2)
PART TIME (REGULAR HOURS) 19 (5.6) 16 (6.9) 3 (2.8)
RETIRED/DISABILITY 7 (2.0) 3 (1.3) 4 (3.8)
STUDENT 11 (3.2) 9 (3.9) 2 (1.9)
UNEMPLOYED 58 (17.0) 38 (16.3) 18 (17.0) 2 (66.7)
EMPLOYMENT_PATTERN_PAST_30_DAYS, n (%)
FULL TIME (35+ HRS/WK) 0 118 (34.4) 73 (31.3) 44 (41.1) 1 (33.3)
HOMEMAKER 12 (3.5) 8 (3.4) 4 (3.7)
IN CONTROLLED ENVIRONMENT 1 (0.3) 1 (0.4)
PART TIME (IRREGULAR DAYWORK) 36 (10.5) 28 (12.0) 8 (7.5)
PART TIME (REGULAR HOURS) 15 (4.4) 11 (4.7) 4 (3.7)
RETIRED/DISABILITY 9 (2.6) 3 (1.3) 6 (5.6)
STUDENT 6 (1.7) 4 (1.7) 2 (1.9)
UNEMPLOYED 146 (42.6) 105 (45.1) 39 (36.4) 2 (66.7)
AGE, mean (SD) 3 37.4 (10.1) 36.9 (10.5) 38.6 (9.3) nan (nan)
DIASTOLIC_BLOOD_PRESSURE, mean (SD) 3 83.7 (40.1) 82.6 (44.3) 86.2 (29.9) 80.0 (9.5)
PULSE, mean (SD) 1 155.3 (32.4) 153.7 (28.7) 158.7 (39.0) 150.7 (42.4)
RESPIRATIONS, mean (SD) 8 17.1 (3.6) 17.1 (3.1) 17.3 (4.5) 16.0 (0.0)
SYSTOLIC_BLOOD_PRESSURE, mean (SD) 3 126.5 (32.5) 123.1 (24.8) 134.0 (44.3) 118.7 (15.9)
TEMPERATURE, mean (SD) 8 103.1 (51.1) 103.2 (60.5) 103.0 (21.1) 98.5 (1.4)
BMI, mean (SD) 6 25.1 (4.9) 24.8 (5.1) 25.7 (4.6) 26.5 (5.6)
METHADONE, n (%)
NEGATIVE 0 56 (16.3) 38 (16.3) 18 (16.8)
NEGATIVENEGATIVE 253 (73.8) 172 (73.8) 78 (72.9) 3 (100.0)
NEGATIVEPOSITIVE 3 (0.9) 2 (0.9) 1 (0.9)
POSITIVE 2 (0.6) 1 (0.4)
POSITIVENEGATIVE 24 (7.0) 16 (6.9) 8 (7.5)
POSITIVEPOSITIVE 5 (1.5) 3 (1.3) 2 (1.9)
MORPHINE, n (%)
NEGATIVE 1 96 (28.1) 67 (28.9) 29 (27.1)
POSITIVE 246 (71.9) 165 (71.1) 78 (72.9) 3 (100.0)
OPIATE, n (%)
NEGATIVE 57 48 (16.8) 36 (18.7) 12 (13.3)
POSITIVE 238 (83.2) 157 (81.3) 78 (86.7) 3 (100.0)
SEX, n (%)
F 0 111 (32.4) 72 (30.9) 38 (35.5) 1 (33.3)
M 231 (67.3) 161 (69.1) 69 (64.5) 1 (33.3)
U 1 (0.3) 1 (33.3)

In [ ]:
# Murat Simsek et Yuquan Dai

In [ ]:
```