Dataset Sources

Beat Acute Myeloid Leukemia (AML) 1.0 was accessed on 13Mar2023 from https://registry.opendata.aws/beataml. OHSU BeatAML Datasets Link: https://ctd2-data.nci.nih.gov/Public/OHSU-1/BeatAML_Waves1_2/

OpenCell Datasets Link: https://opencell.czbiohub.org/download

Check Pre-requisites from the 01-setup Folder

In [4]:
```
%store -r setup_instance_check_passed
```

In [5]:
```
try:
    setup_instance_check_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Instance Check.")
    print("++++++++++++++++++++++++++++++++")
```

In [6]:
```
print(setup_instance_check_passed)
```

```
True
```

In [7]:
```
%store -r setup_dependencies_passed
```

In [8]:
```
try:
    setup_dependencies_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup Dependencies.")
    print("++++++++++++++++++++++++++++++++")
```

In [9]:
```
print(setup_dependencies_passed)
```

```
True
```

In [10]:
```
%store -r setup_s3_bucket_passed
```

In [11]:
```
try:
    setup_s3_bucket_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup S3 Bucket.")
    print("++++++++++++++++++++++++++++++++")
```

In [12]:
```
print(setup_s3_bucket_passed)
```

```
True
```

In [13]:
```
%store -r setup_iam_roles_passed
```

In [14]:
```
try:
    setup_iam_roles_passed
except NameError:
    print("++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup IAM Roles.")
    print("++++++++++++++++++++++++++++++++")
```

In [15]:
```
print(setup_iam_roles_passed)
```

```
True
```

In [16]:
```
if not setup_instance_check_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Instance Check.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_dependencies_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup Dependencies.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_s3_bucket_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup S3 Bucket.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
if not setup_iam_roles_passed:
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
    print("[ERROR] YOU HAVE TO RUN ALL NOTEBOOKS IN THE SETUP FOLDER FIRST. You are missing Setup IAM Roles.")
    print("++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++")
```

In [17]:
```
import boto3
import sagemaker
import pandas as pd
import time
from time import gmtime, strftime

sess = sagemaker.Session()
role = sagemaker.get_execution_role()
bucket = sess.default_bucket()
region = boto3.Session().region_name
account_id = boto3.client("sts").get_caller_identity().get("Account")

sm = boto3.Session().client(service_name="sagemaker", region_name=region)
```

Data Cleaning

Import Tools:

In [127…
```
!pip install klib
```

```
Requirement already satisfied: klib in /opt/conda/lib/python3.7/site-packages (1.0.1)
Requirement already satisfied: Jinja2<4.0.0,>=3.0.3 in /opt/conda/lib/python3.7/site-packages (from klib) (3.1.2)
Requirement already satisfied: matplotlib<4.0.0,>=3.0.3 in /opt/conda/lib/python3.7/site-packages (from klib) (3.1.
3)
Requirement already satisfied: numpy<2.0.0,>=1.16.3 in /opt/conda/lib/python3.7/site-packages (from klib) (1.18.1)
Requirement already satisfied: pandas<2.0.0,>=1.1.2 in /opt/conda/lib/python3.7/site-packages (from klib) (1.3.5)
Requirement already satisfied: scipy<2.0.0,>=1.1.0 in /opt/conda/lib/python3.7/site-packages (from klib) (1.4.1)
Requirement already satisfied: seaborn<0.12.0,>=0.11.1 in /opt/conda/lib/python3.7/site-packages (from klib) (0.11.
2)
Requirement already satisfied: MarkupSafe>=2.0 in /opt/conda/lib/python3.7/site-packages (from Jinja2<4.0.0,>=3.0.3-
>klib) (2.1.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.0,>=3.0.3
->klib) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.0,>=
3.0.3->klib) (1.1.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /opt/conda/lib/python3.7/site-packages (f
rom matplotlib<4.0.0,>=3.0.3->klib) (2.4.6)
Requirement already satisfied: python-dateutil>=2.1 in /opt/conda/lib/python3.7/site-packages (from matplotlib<4.0.
0,>=3.0.3->klib) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.7/site-packages (from pandas<2.0.0,>=1.1.2->kl
ib) (2019.3)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages (from cycler>=0.10->matplotlib<4.0.0,>=
3.0.3->klib) (1.14.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-packages (from kiwisolver>=1.0.1->matplot
lib<4.0.0,>=3.0.3->klib) (45.2.0.post20200210)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system p
ackage manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
```

In [42]:
```
import numpy as np
import seaborn as sns
import klib
import matplotlib.pyplot as plt

%matplotlib inline
%config InlineBackend.figure_format='retina'
```

BeatAML Clinical Summary

OHSU BeatAML Clinical Summary Table

Download & Analyze data sets

In [44]:
```
import numpy as np
import seaborn as sns
import klib
import matplotlib.pyplot as plt
import boto3
import pandas as pd
import matplotlib.pyplot as plt
import json
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
%config InlineBackend.figure_format='retina'
```

In [45]:
```
!aws s3 cp 's3://ads508rawdatasets/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv' ./data/
```

```
download: s3://ads508rawdatasets/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv to data/OHSU_BeatAMLWaves1_2_Tyner_C
linicalSummary.csv
```

In [46]:
```
!aws s3 cp 's3://ads508rawdatasets/opencell-protein-interactions.csv' ./data/
```

```
download: s3://ads508rawdatasets/opencell-protein-interactions.csv to data/opencell-protein-interactions.csv
```

In [47]:
```
import csv

clsm = pd.read_csv('s3://ads508rawdatasets/OHSU_BeatAMLWaves1_2_Tyner_ClinicalSummary.csv')
clsm.head(5)
```

Out[47]:

	LabId	PatientId	consensus_sex	inferred_sex	inferred_ethnicity	centerID	CEBPA_Biallelic	ageAtDiagnosis	isRelapse	isDenovo	...
0	09-00705	163	Male	Male	White	1	n	73.0	False	True	...
1	10-00136	174	Male	Male	White	1	n	69.0	False	True	...
2	10-00172	175	Female	Male	White	1	n	59.0	False	True	...
3	10-00507	45	Female	Female	White	1	n	70.0	False	True	...
4	10-00542	174	Male	Male	White	1	n	69.0	True	False	...

5 rows × 159 columns

In [48]:
```
pi = pd.read_csv('s3://ads508rawdatasets/opencell-protein-interactions.csv')
pi.head(5)
```

Out[48]:

	target_gene_name	interactor_gene_name	target_ensg_id	interactor_ensg_id	interactor_uniprot_ids	
0	AAMP	ARGLU1	ENSG00000127837	ENSG00000134884	Q9NWB6;Q9NWB6-3;Q9NWB6-2	5
1	AAMP	CWF19L2	ENSG00000127837	ENSG00000152404	Q2TBE0;Q2TBE0-2;H7C3G7;Q2TBE0-3;H0YE03	5
2	AAMP	PRPF40A	ENSG00000127837	ENSG00000196504	A0A3F2YNY6;O75400-2;O75400-3;O75400;H0YG38;F5H578	5
3	AAMP	RPL10	ENSG00000127837	ENSG00000147403	X1WI28;P27635;B8A6G2;A6QRI9;Q96L21	15
4	AAMP	RSRC1	ENSG00000127837	ENSG00000174891	Q96IZ7-2;Q96IZ7;H7C5Q0;C9J713;C9J367;C9J8Q2;C9...	5

In [ ]:
```
```

Display clsm data set

In [49]:
```
clsm
```

Out[49]:

	LabId	PatientId	consensus_sex	inferred_sex	inferred_ethnicity	centerID	CEBPA_Biallelic	ageAtDiagnosis	isRelapse	isDenovo	..
0	09-00705	163	Male	Male	White	1	n	73.0	False	True	.
1	10-00136	174	Male	Male	White	1	n	69.0	False	True	.
2	10-00172	175	Female	Male	White	1	n	59.0	False	True	.
3	10-00507	45	Female	Female	White	1	n	70.0	False	True	.
4	10-00542	174	Male	Male	White	1	n	69.0	True	False	.
...	...	...	...	...	...	...	...	...	...	...	.
667	17-00072	4366	Male	Male	White	1	n	70.0	False	False	.
668	17-00077	4317	Female	Female	White	1	n	72.0	False	False	.
669	17-00093	4379	Female	Female	Black	2	n	43.0	False	False	.
670	17-00094	4380	Male	Male	White	6	n	57.0	False	False	.
671	17-00096	2747	Male	Male	White	6	n	62.0	False	False	.

672 rows × 159 columns

In [50]:
```
clsm.shape
```

Out[50]:
```
(672, 159)
```

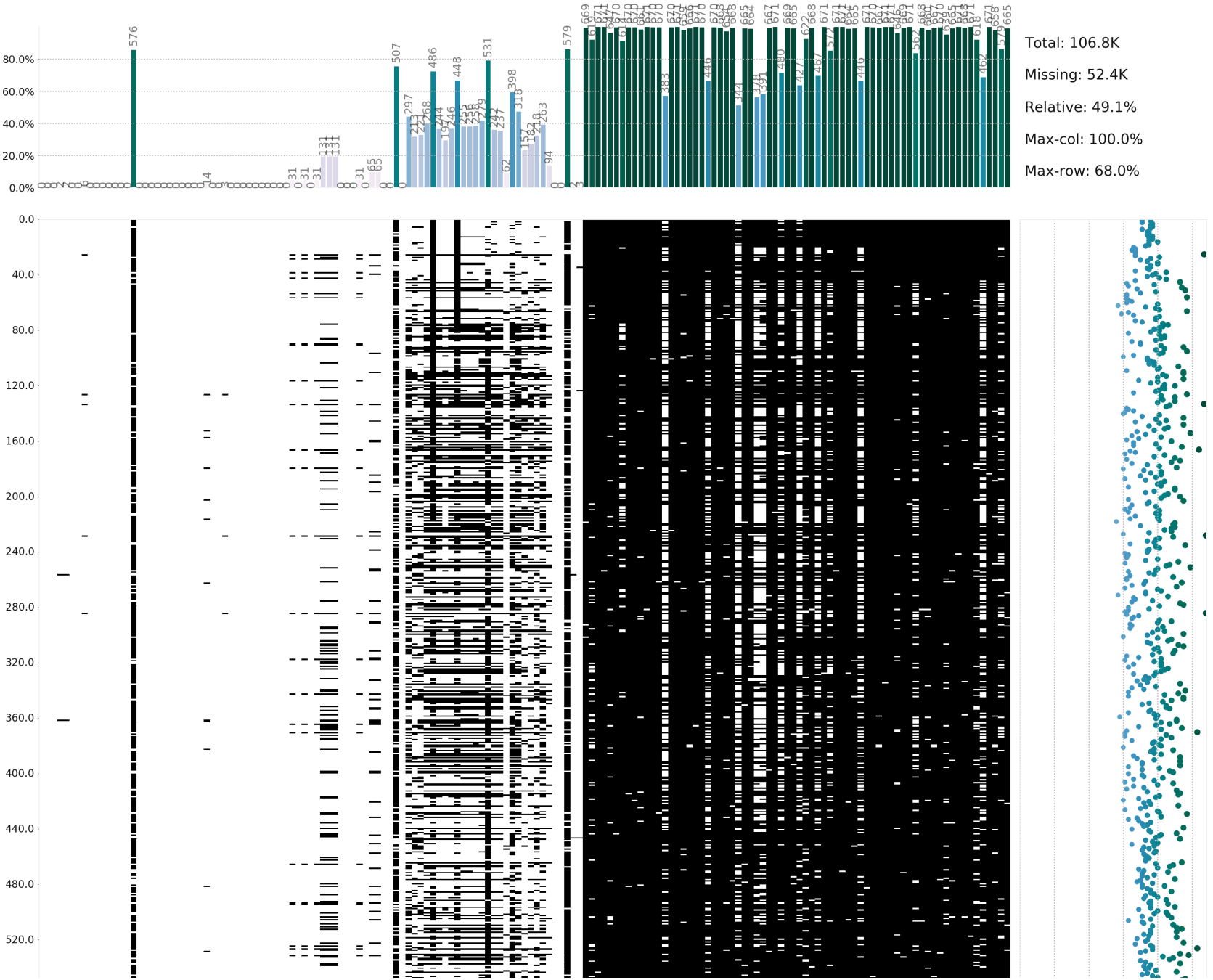
In [51]:
```
clsm = clsm.replace('', np.NAN)
clsm.info()
```

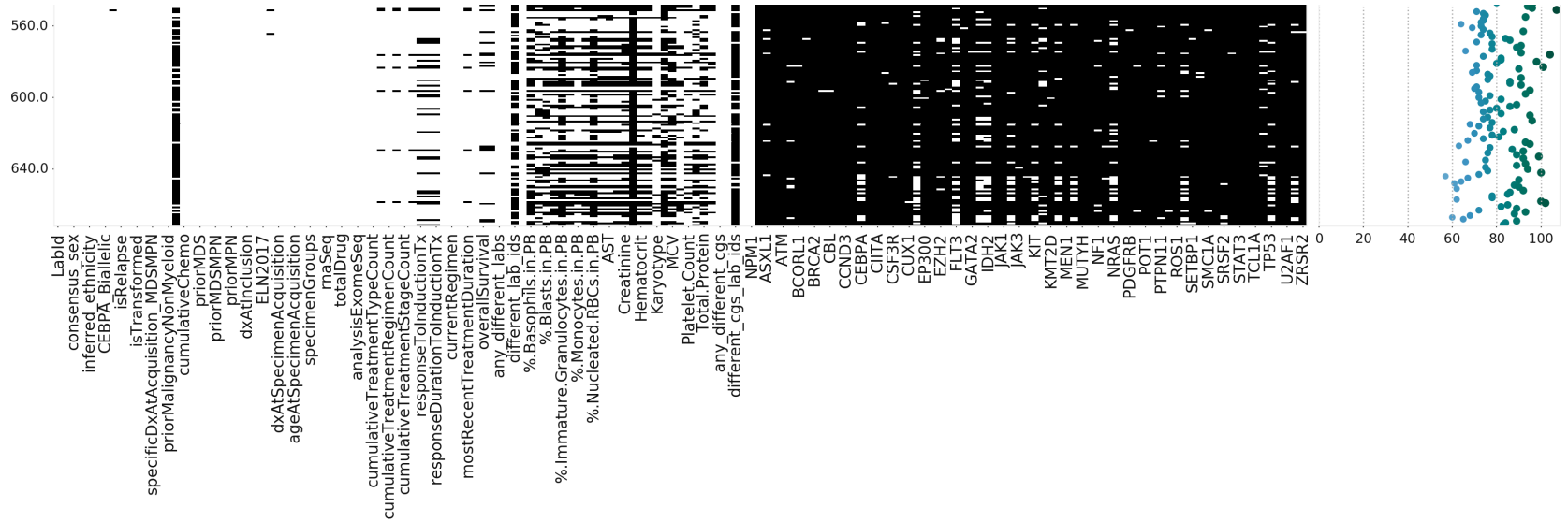
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Columns: 159 entries, LabId to ZRSR2
dtypes: bool(9), float64(22), int64(7), object(121)
memory usage: 793.5+ KB
```

In [52]:
```
klib.missingval_plot(clsm)
```

Out[52]:
```
GridSpec(6, 6)
```

Missing value plot





Create new dataframe to retain relevant features for further use

In [53]:
```
clsm_cut = pd.DataFrame(clsm[['LabId', 'PatientId', 'consensus_sex', 'inferred_ethnicity', 'isRelapse',
                              'isTransformed', 'priorMalignancyNonMyeloid', 'priorMDS', 'priorMDSMPN', 'priorMPN',
                              'ELN2017', 'dxAtSpecimenAcquisition', 'vitalStatus', 'overallSurvival', '%.Blasts.in.BM
                               '%.Blasts.in.PB', 'FLT3-ITD', 'NPM1']])

clsm_cut
```

Out[53]:

	LabId	PatientId	consensus_sex	inferred_ethnicity	isRelapse	isTransformed	priorMalignancyNonMyeloid	priorMDS	priorMDSMP
0	09-00705	163	Male	White	False	False	n	n	
1	10-00136	174	Male	White	False	False	n	n	
2	10-00172	175	Female	White	False	False	n	n	
3	10-00507	45	Female	White	False	False	n	n	
4	10-00542	174	Male	White	True	False	n	n	
...	...	...	...	...	...	...	...	...	...
667	17-00072	4366	Male	White	False	True	n	n	
668	17-00077	4317	Female	White	False	False	n	n	
669	17-00093	4379	Female	Black	False	True	n	n	
670	17-00094	4380	Male	White	False	True	n	n	
671	17-00096	2747	Male	White	False	True	n	n	

672 rows × 18 columns

In [54]:
```
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   LabId                       672 non-null    object
 1   PatientId                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          670 non-null    object
 4   isRelapse                   672 non-null    bool
 5   isTransformed               672 non-null    bool
 6   priorMalignancyNonMyeloid   672 non-null    object
 7   priorMDS                    672 non-null    object
 8   priorMDSMPN                 672 non-null    object
 9   priorMPN                    672 non-null    object
 10  ELN2017                     672 non-null    object
 11  dxAtSpecimenAcquisition     672 non-null    object
 12  vitalStatus                 672 non-null    object
 13  overallSurvival             607 non-null    float64
 14  %.Blasts.in.BM              459 non-null    object
 15  %.Blasts.in.PB              451 non-null    object
 16  FLT3-ITD                    670 non-null    object
 17  NPM1                        669 non-null    object
dtypes: bool(2), float64(1), int64(1), object(14)
memory usage: 85.4+ KB
```

In [55]:
```
clsm_cut.describe()
```

Out[55]:

	PatientId	overallSurvival
count	672.000000	607.000000
mean	2088.020833	441.881384
std	973.372734	479.180429
min	17.000000	-1.000000
25%	1450.750000	167.000000
50%	2016.000000	323.000000
75%	2501.500000	555.000000
max	4380.000000	5305.000000

Attribute Information

% Blasts Attributes Numerical Prep

%.blasts.in.bm Attribute:

In [56]:
```
#Attribute Transformation - %.Blasts.in.BM'
#Identify unique values in %.Blasts.in.BM'
clsm_cut['%.Blasts.in.BM'].unique()
```

Out[56]:
```
array(['94', '80', '91', '97', '87', nan, '40', '75', '83', '95', '85',
       '90', '70', '92', '72', '68', '88', '36', '81', '93', '34', '77.5',
       '46', '65', '50', '76', '71', '60', '73', '55', '0.5', '30', '62',
       '18', '82', '28', '41', '64', '84', '21', '51', '17', '49.4', '32',
       '29', '25', '59.3', '66', '20', '52', '54', '22', '10', '12', '13',
       '67', '39', '25.9', '45', '37', '78', '8', '3', '54.8', '74', '96',
       '4', '86.1', '42', '56', '69', '79', '33', '9', '0.4', '51.5',
       '15', '5', '24', '7', '2', '6', '1', '58', '>50', '35', '86',
       '93.2', '0', '27', '89.6', '23', '98', '19', '91.8', '>95', '57',
       '71.5', '78.3', '63', '1.5', '53.74', '59.5', '44', '42.5', '26',
       '3.5', '48', '26.3', '47', '88.5'], dtype=object)
```

In [57]:
```
# > and < will be changed to whole numbers less than or greater than.
clsm_cut['%.Blasts.in.BM'] = clsm_cut['%.Blasts.in.BM'].replace(['>50'], 51)
clsm_cut['%.Blasts.in.BM'] = clsm_cut['%.Blasts.in.BM'].replace(['>95'], 96)

clsm_cut['%.Blasts.in.BM'].unique()
```

Out[57]:
```
array(['94', '80', '91', '97', '87', nan, '40', '75', '83', '95', '85',
       '90', '70', '92', '72', '68', '88', '36', '81', '93', '34', '77.5',
       '46', '65', '50', '76', '71', '60', '73', '55', '0.5', '30', '62',
       '18', '82', '28', '41', '64', '84', '21', '51', '17', '49.4', '32',
       '29', '25', '59.3', '66', '20', '52', '54', '22', '10', '12', '13',
       '67', '39', '25.9', '45', '37', '78', '8', '3', '54.8', '74', '96',
       '4', '86.1', '42', '56', '69', '79', '33', '9', '0.4', '51.5',
       '15', '5', '24', '7', '2', '6', '1', '58', 51, '35', '86', '93.2',
       '0', '27', '89.6', '23', '98', '19', '91.8', 96, '57', '71.5',
       '78.3', '63', '1.5', '53.74', '59.5', '44', '42.5', '26', '3.5',
       '48', '26.3', '47', '88.5'], dtype=object)
```

In [58]:
```
#Attribute Transformation - %.Blasts.in.PB'
#Identify unique values in %.Blasts.in.PB'
clsm_cut['%.Blasts.in.PB'].unique()
```

Out[58]:
```
array(['97', '19', '99', '80', nan, '51', '30', '41', '84', '77', '75',
       '63', '60', '96', '66', '45', '93', '9', '82', '15', '33', '0',
       '13', '94', '89', '83', '>90', '78', '72', '59', '32', '6', '29',
       '24', '64', '57', '52', '2.1', '<5', '17', '22', '5', '47', '56',
       '25', '23', '42', '65', '71', '8', '3.5', '66.3', '95', '44', '10',
       '28.6', '18', '58', '67', '40', '92', '54', '1', '2', '20', '28',
       '35', '85', '42.4', '16', '49.1', '14', '88', '46', '7', '0.5',
       '79', '26', '87', '20.4', '68', '48', '5.3', '61', '90', '17.4',
       '57.4', '43.8', '50', '37', '4', '3', '12', '81', '11', '90.5',
       '"rare"', '90.2', '55', 'rare', '39', '31', '86', '47.4', '27.4',
       '39.6', '12.9', '15.4', '9.5', '62', '64.6', '27.8', '69.14',
       '52.2', '91', '67.25', '49', '23.7', '48.6', '98', '74.8', '2.6',
       '43', '29.6', '47.5', '38', '2.5', '25.2', '3.56', '70', '99.2',
       '73', '26.7', '38.5', '7.7', '74', '93.3', '12.1', '11.2', '92.9',
       '98.4', '6.8', '10.5', '53', '3.1', '28.9', '72.9', '40.2', '3.3',
       '42.1', '11.5', '77.8', '3.8', '59.5', '21.7', '53.2'],
      dtype=object)
```

In [59]:
```
#%.Blasts.in.PB attribute has 1 "rare" and 1 'rare' record with no flt3 nor npm1 input. This will be changed to NAN
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].replace(['"rare"'], np.nan)
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].replace(['rare'], np.nan)
# > and < will be changed to whole numbers less than or greater than.
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].replace(['<5'], 4)
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].replace(['>90'], 91)

clsm_cut['%.Blasts.in.PB'].unique()
```

Out[59]:
```
array(['97', '19', '99', '80', nan, '51', '30', '41', '84', '77', '75',
       '63', '60', '96', '66', '45', '93', '9', '82', '15', '33', '0',
       '13', '94', '89', '83', 91, '78', '72', '59', '32', '6', '29',
       '24', '64', '57', '52', '2.1', 4, '17', '22', '5', '47', '56',
       '25', '23', '42', '65', '71', '8', '3.5', '66.3', '95', '44', '10',
       '28.6', '18', '58', '67', '40', '92', '54', '1', '2', '20', '28',
       '35', '85', '42.4', '16', '49.1', '14', '88', '46', '7', '0.5',
       '79', '26', '87', '20.4', '68', '48', '5.3', '61', '90', '17.4',
       '57.4', '43.8', '50', '37', '4', '3', '12', '81', '11', '90.5',
       '90.2', '55', '39', '31', '86', '47.4', '27.4', '39.6', '12.9',
       '15.4', '9.5', '62', '64.6', '27.8', '69.14', '52.2', '91',
       '67.25', '49', '23.7', '48.6', '98', '74.8', '2.6', '43', '29.6',
       '47.5', '38', '2.5', '25.2', '3.56', '70', '99.2', '73', '26.7',
       '38.5', '7.7', '74', '93.3', '12.1', '11.2', '92.9', '98.4', '6.8',
       '10.5', '53', '3.1', '28.9', '72.9', '40.2', '3.3', '42.1', '11.5',
       '77.8', '3.8', '59.5', '21.7', '53.2'], dtype=object)
```

From Categorical to Numerical

Transform &.blasts.in.bm and %.blasts.in.pb from object to float:

In [60]:
```
clsm_cut['%.Blasts.in.BM'] = clsm_cut['%.Blasts.in.BM'].astype(float)
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].astype(float)
```

In [61]:
```
clsm_cut.info()
```

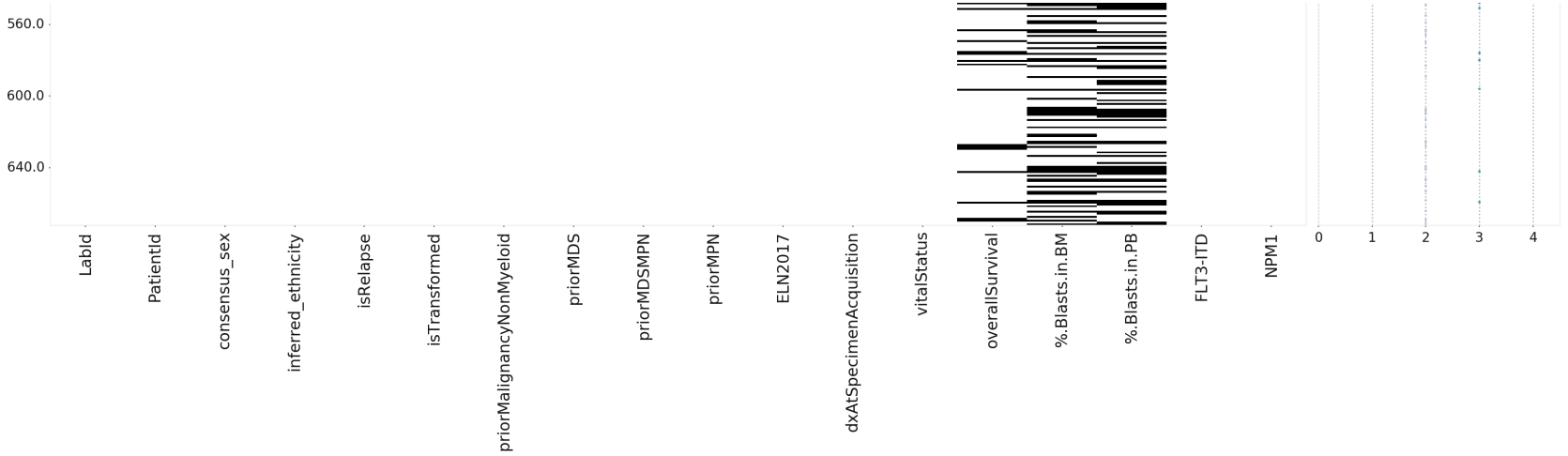
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   LabId                       672 non-null    object
 1   PatientId                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          670 non-null    object
 4   isRelapse                   672 non-null    bool
 5   isTransformed               672 non-null    bool
 6   priorMalignancyNonMyeloid   672 non-null    object
 7   priorMDS                    672 non-null    object
 8   priorMDSMPN                 672 non-null    object
 9   priorMPN                    672 non-null    object
 10  ELN2017                     672 non-null    object
 11  dxAtSpecimenAcquisition     672 non-null    object
 12  vitalStatus                 672 non-null    object
 13  overallSurvival             607 non-null    float64
 14  %.Blasts.in.BM              459 non-null    float64
 15  %.Blasts.in.PB              448 non-null    float64
 16  FLT3-ITD                    670 non-null    object
 17  NPM1                        669 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

clsm_cut Identify Missing Values

In [62]:
```
klib.missingval_plot(clsm_cut)
```

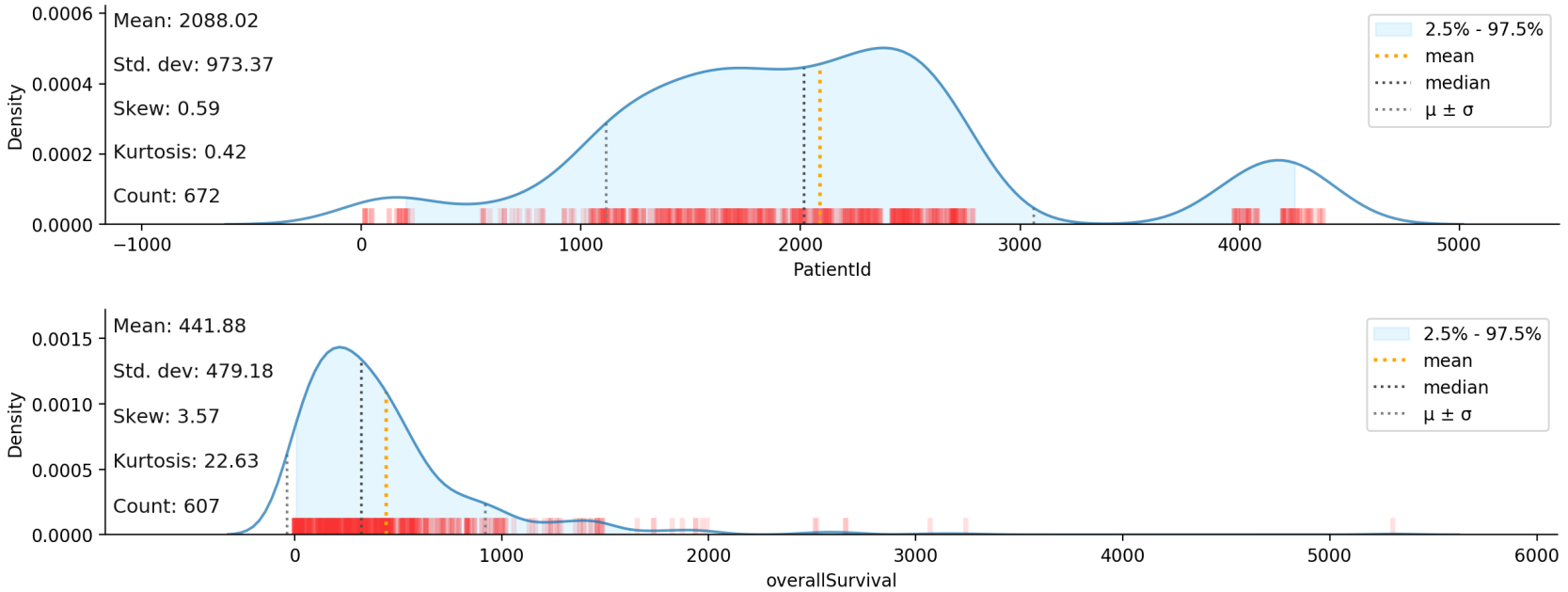
Out[62]:
```
GridSpec(6, 6)
```

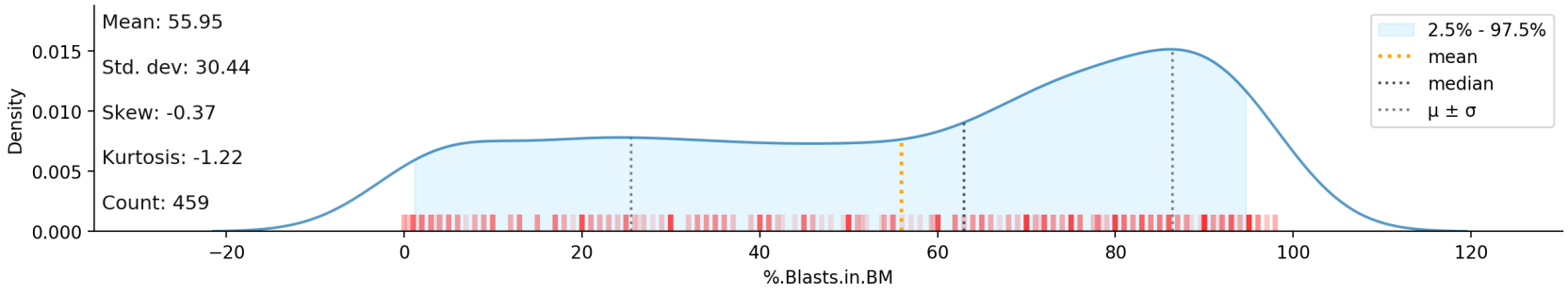
Missing value plot

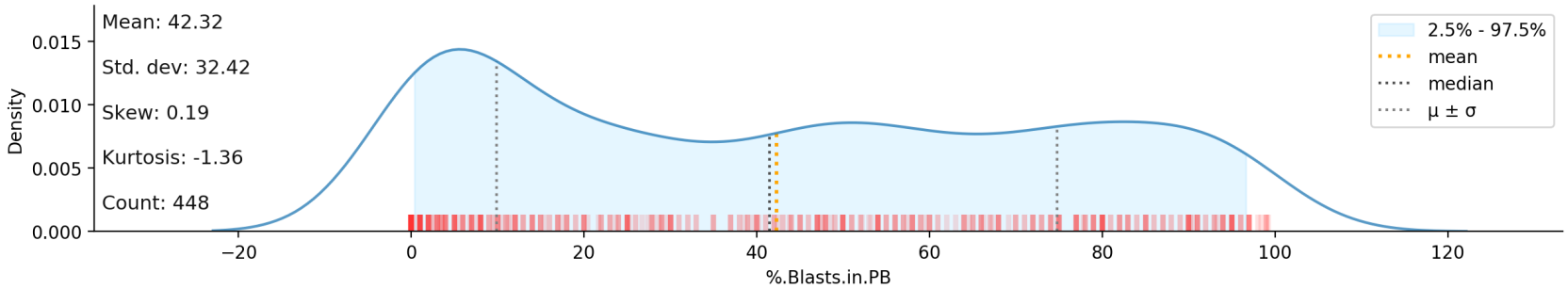


In [63]:
```
#Replace Missing Value
klib.dist_plot(clsm_cut)
```

Out[63]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8a235cd10>
```







In [64]:
```
clsm_cut.describe()
```

Out[64]:

	PatientId	overallSurvival	%.Blasts.in.BM	%.Blasts.in.PB
count	672.000000	607.000000	459.000000	448.000000
mean	2088.020833	441.881384	55.949325	42.316629
std	973.372734	479.180429	30.440925	32.418249
min	17.000000	-1.000000	0.000000	0.000000
25%	1450.750000	167.000000	30.000000	10.000000
50%	2016.000000	323.000000	63.000000	41.500000
75%	2501.500000	555.000000	83.000000	72.000000
max	4380.000000	5305.000000	98.000000	99.200000

In [65]:
```
#From distibution, skewness suggest median is the best representation.
clsm_cut['overallSurvival'] = clsm_cut['overallSurvival'].fillna(clsm_cut['overallSurvival'].median())
clsm_cut['%.Blasts.in.BM'] = clsm_cut['%.Blasts.in.BM'].fillna(clsm_cut['%.Blasts.in.BM'].median())
clsm_cut['%.Blasts.in.PB'] = clsm_cut['%.Blasts.in.PB'].fillna(clsm_cut['%.Blasts.in.PB'].median())
```

In [66]:
```
#Replace categorical NaN with unknown
clsm_cut = clsm_cut.replace(np.nan, 'unknown', regex=True)
```

In [67]:
```
#Determine mode of inferred_ethnicity
clsm_cut['inferred_ethnicity'].mode()
```

Out[67]:
```
0    White
dtype: object
```

In [68]:
```
#In inferred_ethnicity, replace mode of unknown to white:
clsm_cut['inferred_ethnicity'] = clsm_cut['inferred_ethnicity'].replace(['unknown'], 'white')

clsm_cut['inferred_ethnicity'].unique()
```

Out[68]:
```
array(['White', 'HispNative', 'AdmixedBlack', 'Asian', 'Black',
       'AdmixedAsian', 'white', 'AdmixedWhite', 'AdmixedHispNative'],
      dtype=object)
```

In [69]:
```
#Determine mode of flt3-itd
clsm_cut['FLT3-ITD'].mode()
```

Out[69]:
```
0    negative
dtype: object
```

In [70]:
```
#In flt3-itd, replace mode of unknown to negative:
clsm_cut['FLT3-ITD'] = clsm_cut['FLT3-ITD'].replace(['unknown'], 'negative')
clsm_cut['FLT3-ITD'].unique()
```

Out[70]:
```
array(['positive', 'negative'], dtype=object)
```

In [71]:
```
clsm_cut['NPM1'].mode()
```

Out[71]:
```
0    negative
dtype: object
```

In [72]:
```
#In npm1, replace mode of unknown to negative:
clsm_cut['NPM1'] = clsm_cut['NPM1'].replace(['unknown'], 'negative')
clsm_cut['NPM1'].unique()
```

Out[72]:
```
array(['positive', 'negative'], dtype=object)
```

In [73]:
```
#Check for missing values
klib.missingval_plot(clsm_cut)
```

```
No missing values found in the dataset.
```

In [74]:
```
clsm_cut.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   LabId                      672 non-null    object 
 1   PatientId                  672 non-null    int64  
 2   consensus_sex              672 non-null    object 
 3   inferred_ethnicity         672 non-null    object 
 4   isRelapse                  672 non-null    bool   
 5   isTransformed              672 non-null    bool   
 6   priorMalignancyNonMyeloid  672 non-null    object 
 7   priorMDS                   672 non-null    object 
 8   priorMDSMPN                672 non-null    object 
 9   priorMPN                   672 non-null    object 
 10  ELN2017                    672 non-null    object 
 11  dxAtSpecimenAcquisition    672 non-null    object 
 12  vitalStatus                672 non-null    object 
 13  overallSurvival            672 non-null    float64
 14  %.Blasts.in.BM             672 non-null    float64
 15  %.Blasts.in.PB             672 non-null    float64
 16  FLT3-ITD                   672 non-null    object 
 17  NPM1                       672 non-null    object 
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

Check for Duplicates

In [75]:
```
#Remove duplicated columns
clsm_cut = clsm_cut.drop_duplicates(ignore_index=True)
clsm_cut.info()
```

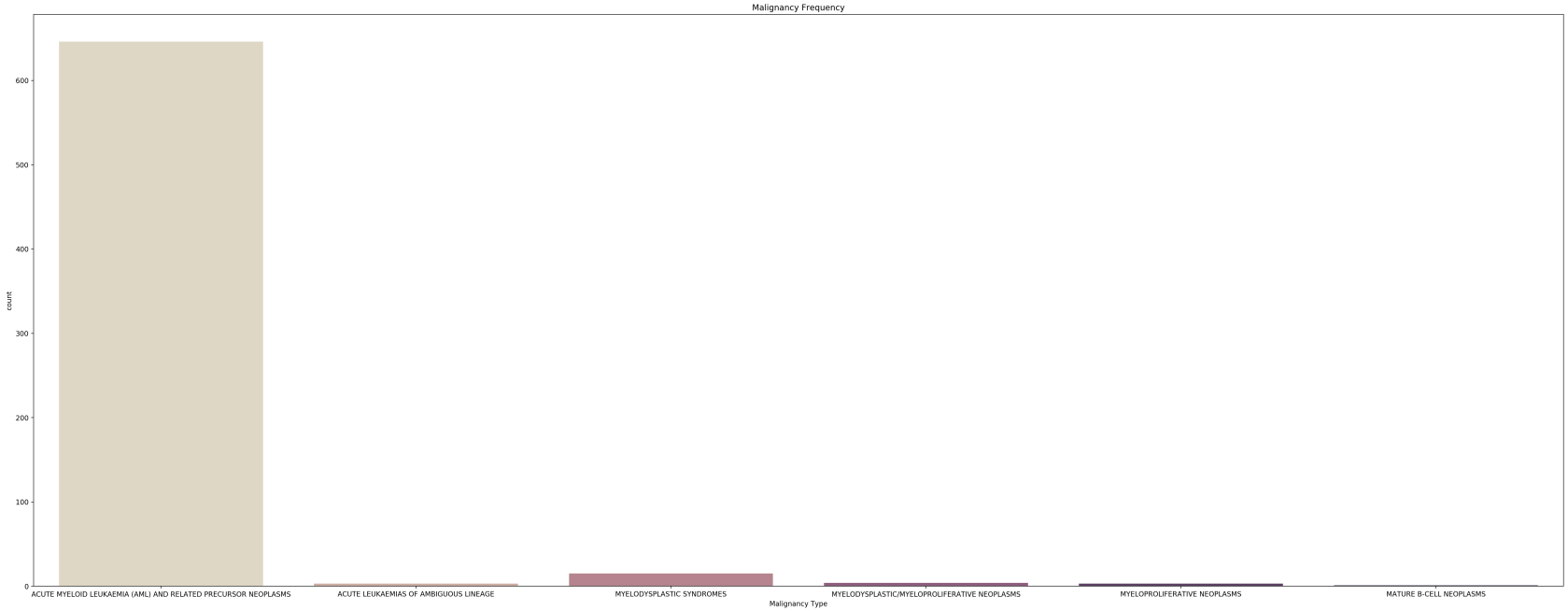
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 18 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   LabId                       672 non-null    object
 1   PatientId                   672 non-null    int64
 2   consensus_sex               672 non-null    object
 3   inferred_ethnicity          672 non-null    object
 4   isRelapse                   672 non-null    bool
 5   isTransformed               672 non-null    bool
 6   priorMalignancyNonMyeloid   672 non-null    object
 7   priorMDS                    672 non-null    object
 8   priorMDSMPN                 672 non-null    object
 9   priorMPN                    672 non-null    object
 10  ELN2017                     672 non-null    object
 11  dxAtSpecimenAcquisition     672 non-null    object
 12  vitalStatus                 672 non-null    object
 13  overallSurvival             672 non-null    float64
 14  %.Blasts.in.BM              672 non-null    float64
 15  %.Blasts.in.PB              672 non-null    float64
 16  FLT3-ITD                    672 non-null    object
 17  NPM1                        672 non-null    object
dtypes: bool(2), float64(3), int64(1), object(12)
memory usage: 85.4+ KB
```

In [76]:
```
clsm_cut.duplicated().sum()
```

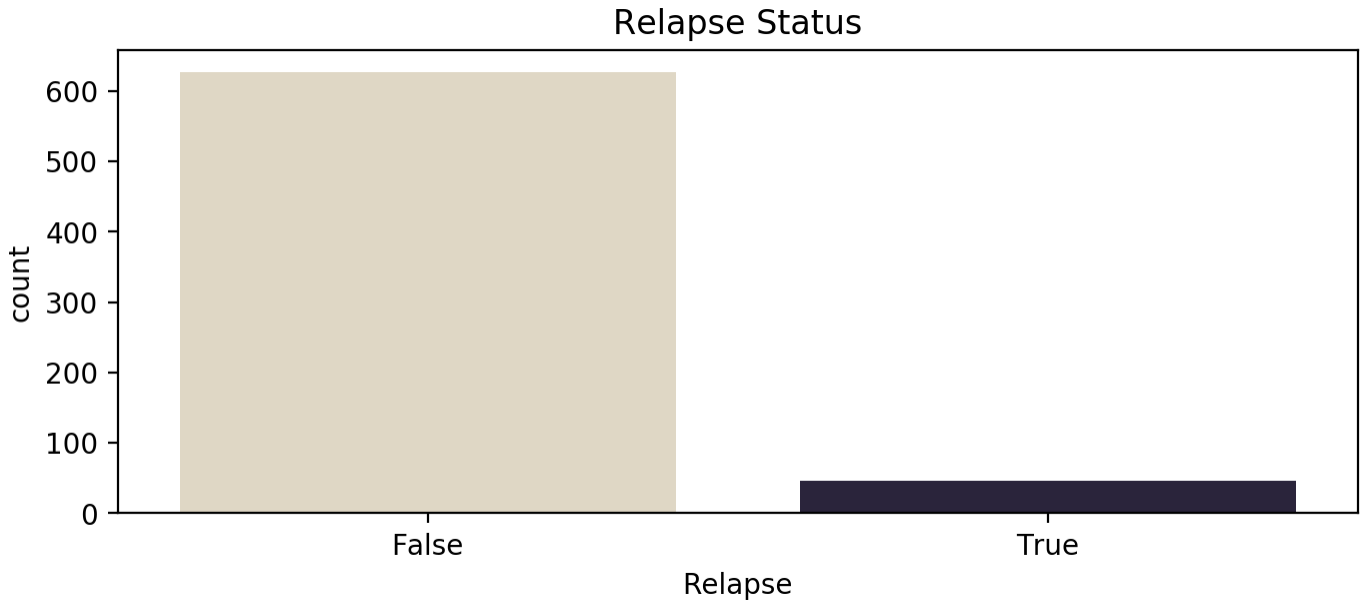
Out[76]:
```
0
```

Data Exploration

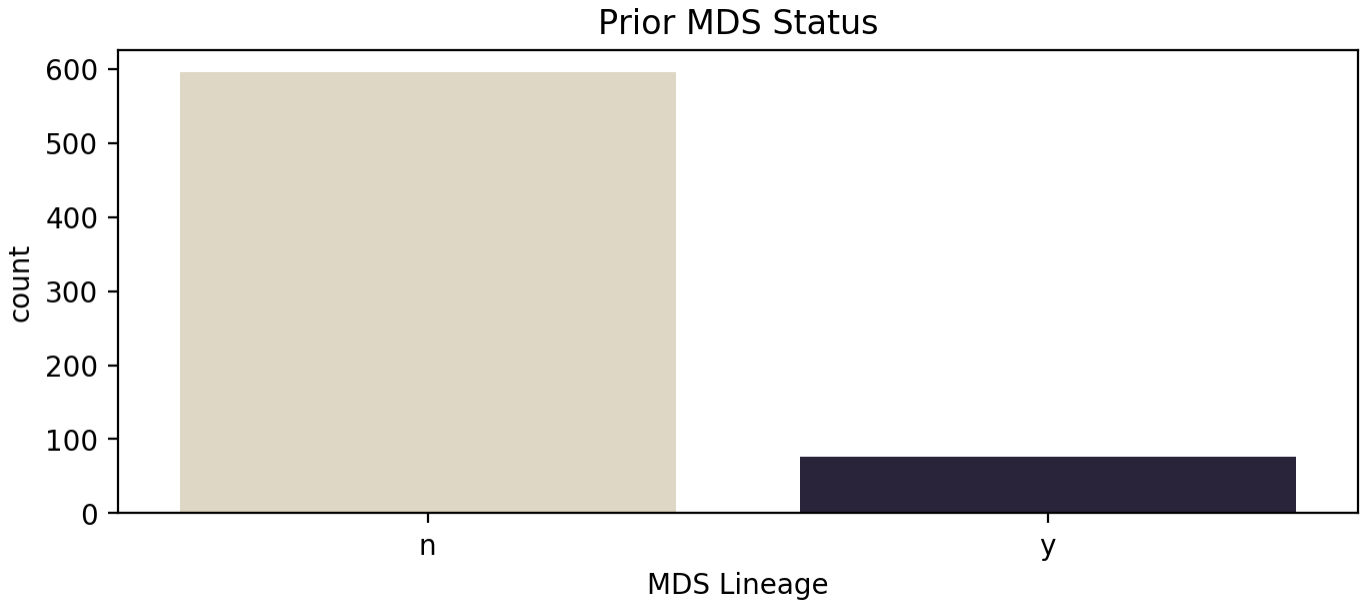
In [77]:
```
#Data Visualization - "dxAtSpecimenAcquisition
sns.countplot(x=clsm_cut["dxAtSpecimenAcquisition"], palette = "ch:s=-.2,r=.6")
plt.xlabel('Malignancy Type')
plt.title('Malignancy Frequency')
plt.gcf().set_size_inches(40, 15)
```



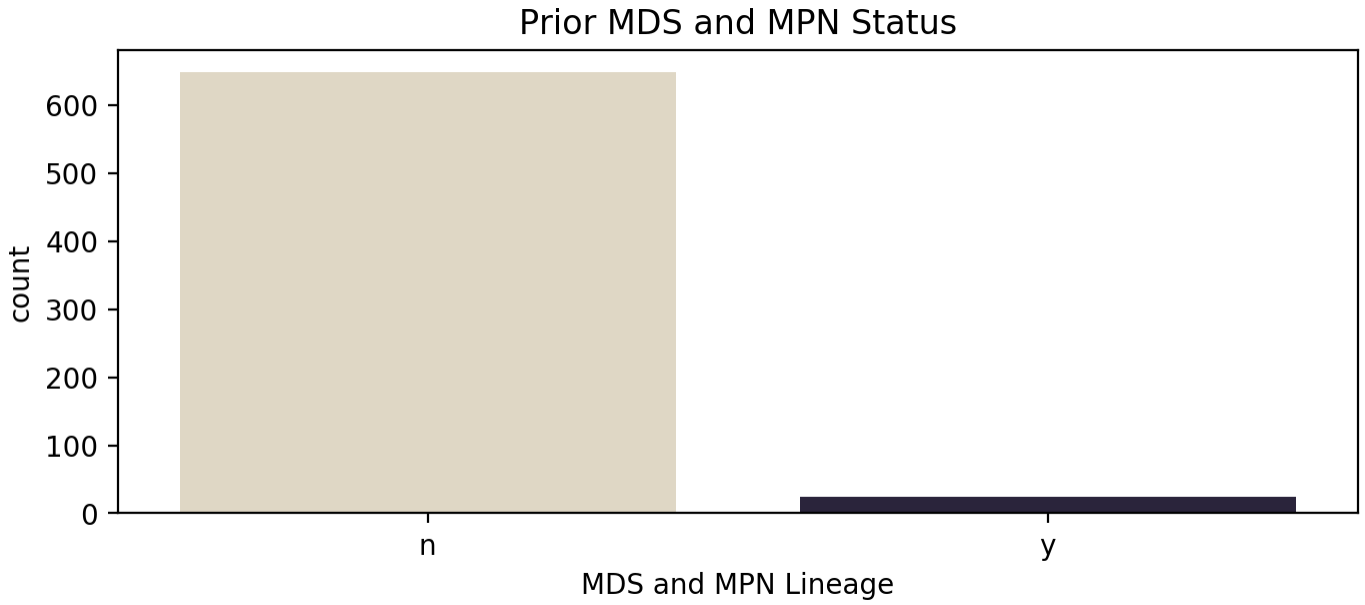
In [78]:
```
#Data Visualization - "isRelapse"
sns.countplot(x=clsm_cut["isRelapse"], palette = "ch:s=-.2,r=.6")
plt.xlabel('Relapse')
plt.title('Relapse Status')
plt.gcf().set_size_inches(8, 3)
```



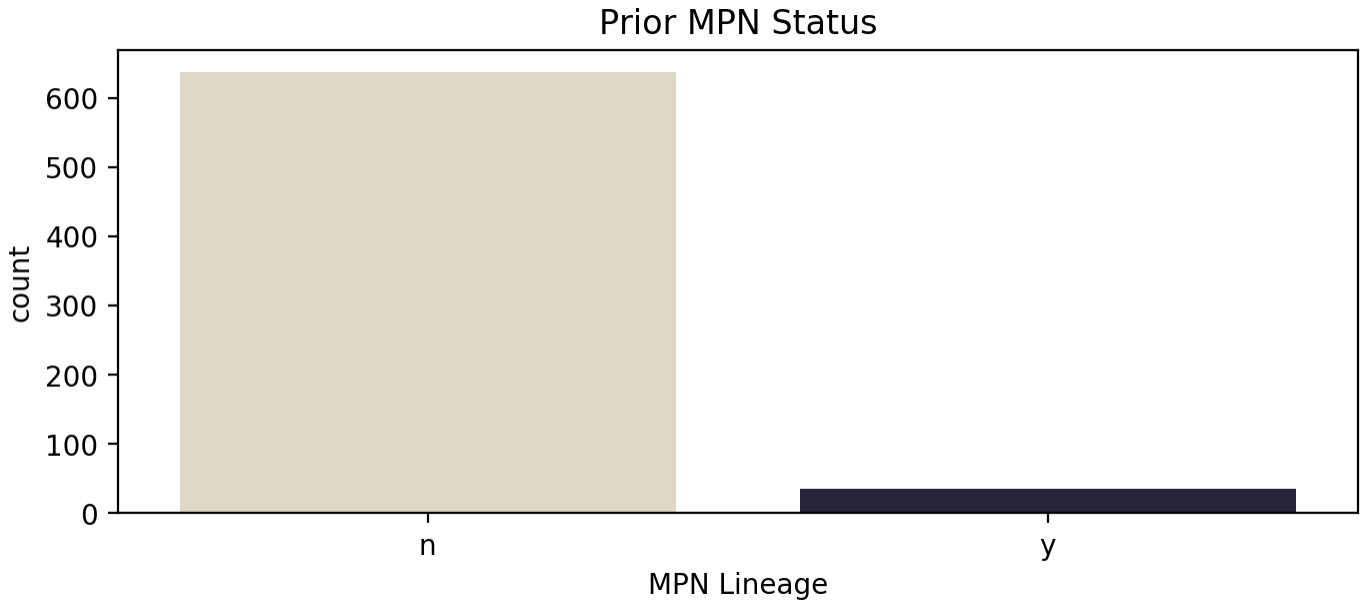
In [79]:
```
#Data Visualization - "priorMDS"
sns.countplot(x=clsm_cut["priorMDS"], palette = "ch:s=-.2,r=.6")
plt.xlabel('MDS Lineage')
plt.title('Prior MDS Status')
plt.gcf().set_size_inches(8, 3)
```



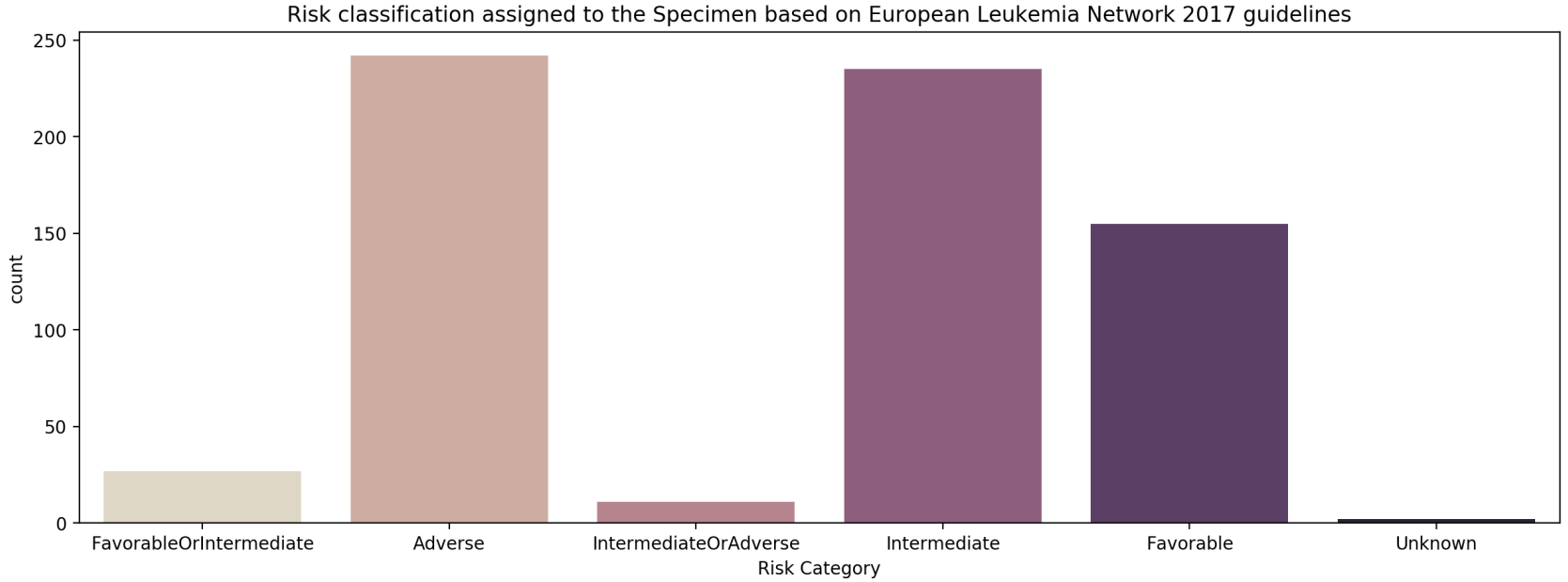
In [80]:
```
#Data Visualization - "priorMDSMPN"
sns.countplot(x=clsm_cut["priorMDSMPN"], palette = "ch:s=-.2,r=.6")
plt.xlabel('MDS and MPN Lineage')
plt.title('Prior MDS and MPN Status')
plt.gcf().set_size_inches(8, 3)
```



In [81]:
```
#Data Visualization - "priorMPN"
sns.countplot(x=clsm_cut["priorMPN"], palette = "ch:s=-.2,r=.6")
plt.xlabel('MPN Lineage')
plt.title('Prior MPN Status')
plt.gcf().set_size_inches(8, 3)
```

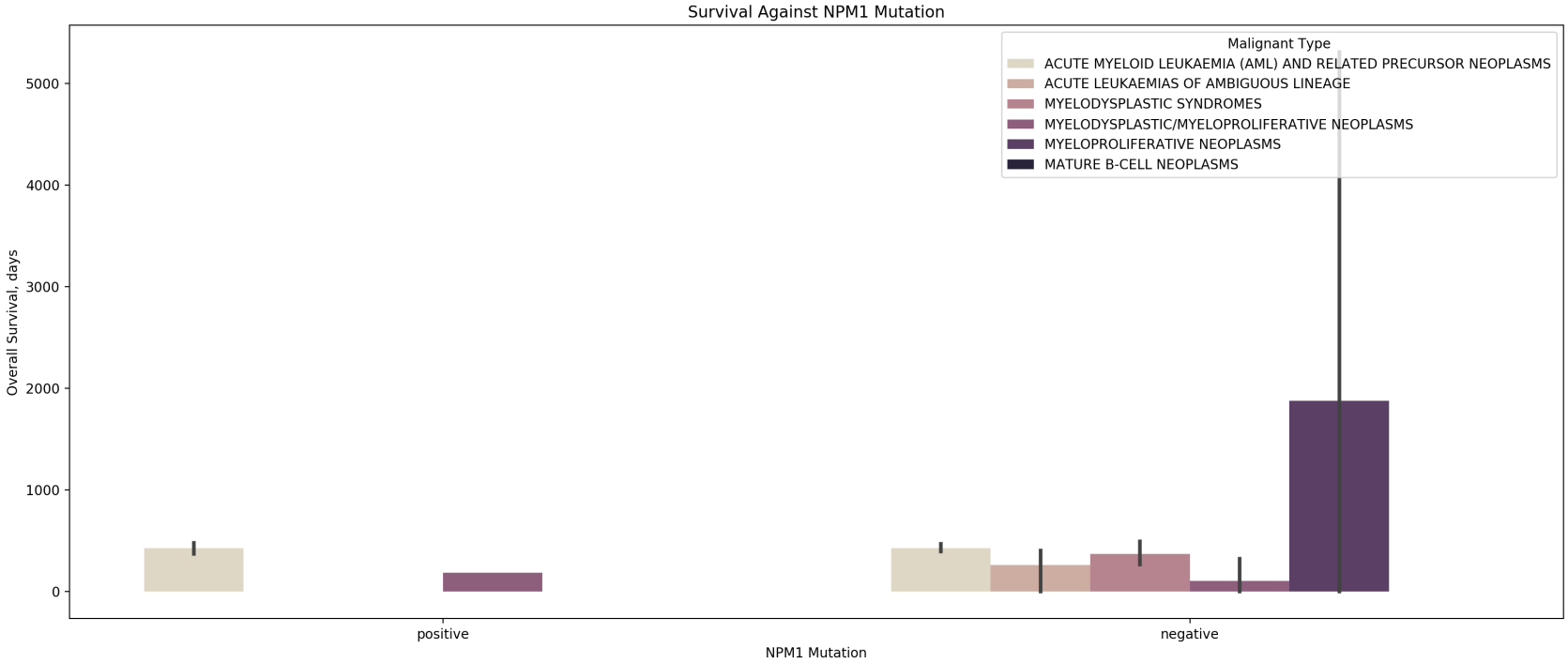


In [82]:
```
#Data Visualization - "ELN2017"
sns.countplot(x=clsm_cut["ELN2017"], palette = "ch:s=-.2,r=.6")
plt.xlabel('Risk Category')
plt.title('Risk classification assigned to the Specimen based on European Leukemia Network 2017 guidelines')
plt.gcf().set_size_inches(15, 5)
```



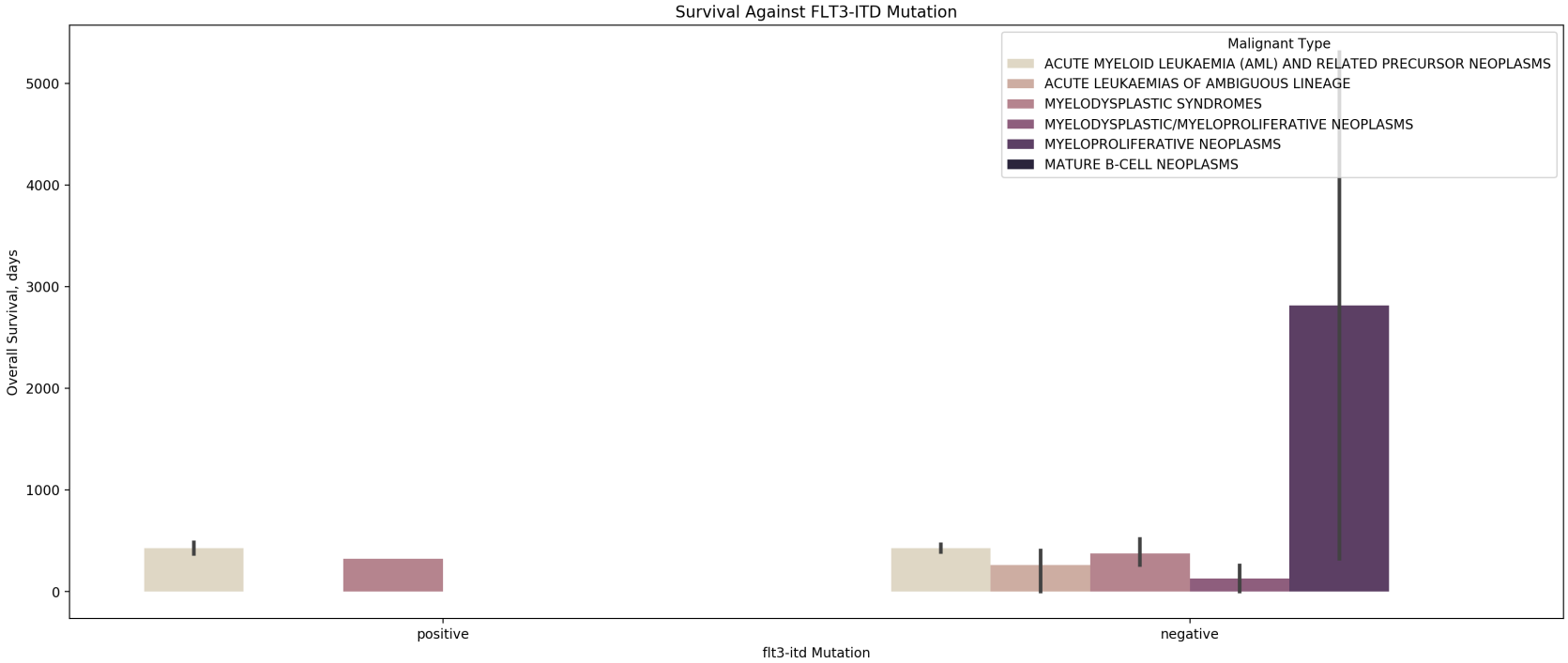
In [83]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against NPM1 Mutation")
```

Out[83]:
```
Text(0.5, 1.0, 'Survival Against NPM1 Mutation')
```



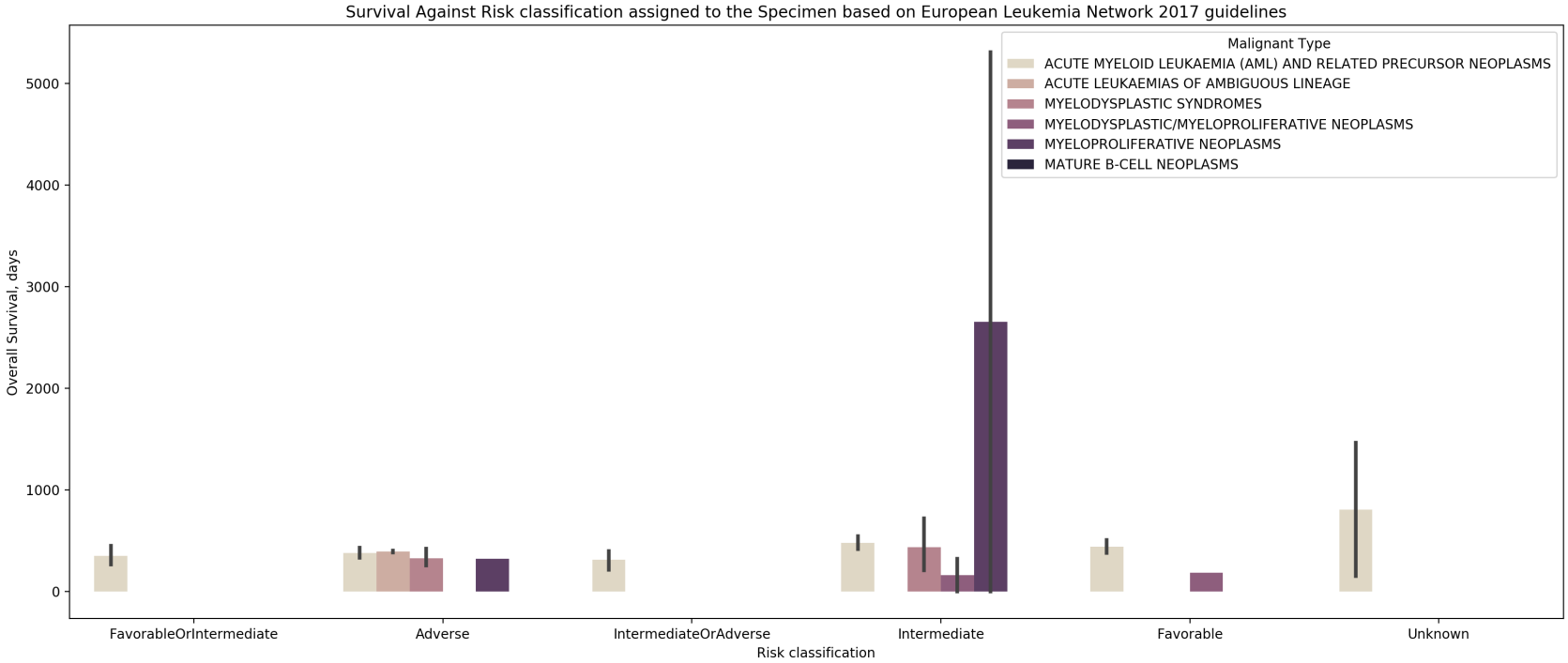
In [84]:
```
sns.barplot(data= clsm_cut,x = 'FLT3-ITD', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('flt3-itd Mutation')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against FLT3-ITD Mutation")
```

Out[84]:
```
Text(0.5, 1.0, 'Survival Against FLT3-ITD Mutation')
```



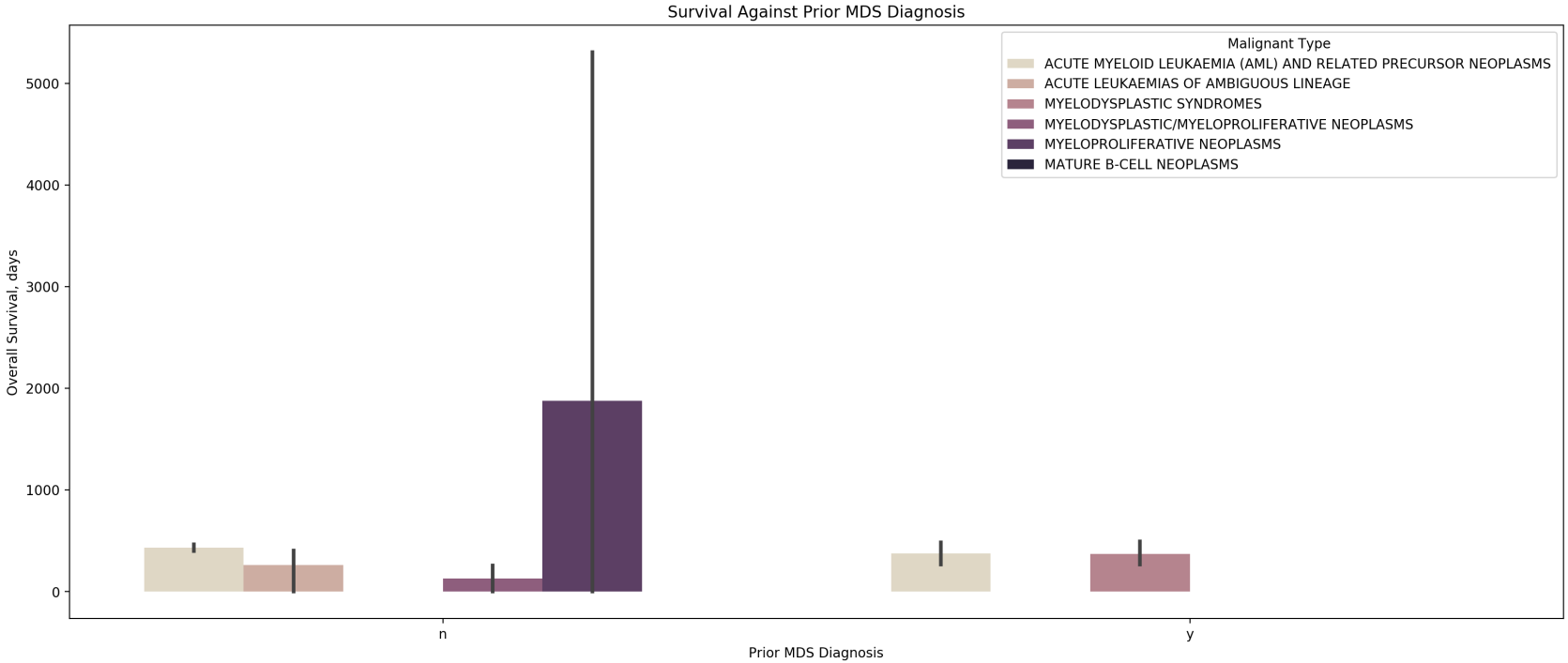
In [85]:
```
sns.barplot(data= clsm_cut,x = 'ELN2017', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('Risk classification')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against Risk classification assigned to the Specimen based on European Leukemia Network 2017 guic
```

Out[85]:
```
Text(0.5, 1.0, 'Survival Against Risk classification assigned to the Specimen based on European Leukemia Network 201
7 guidelines')
```



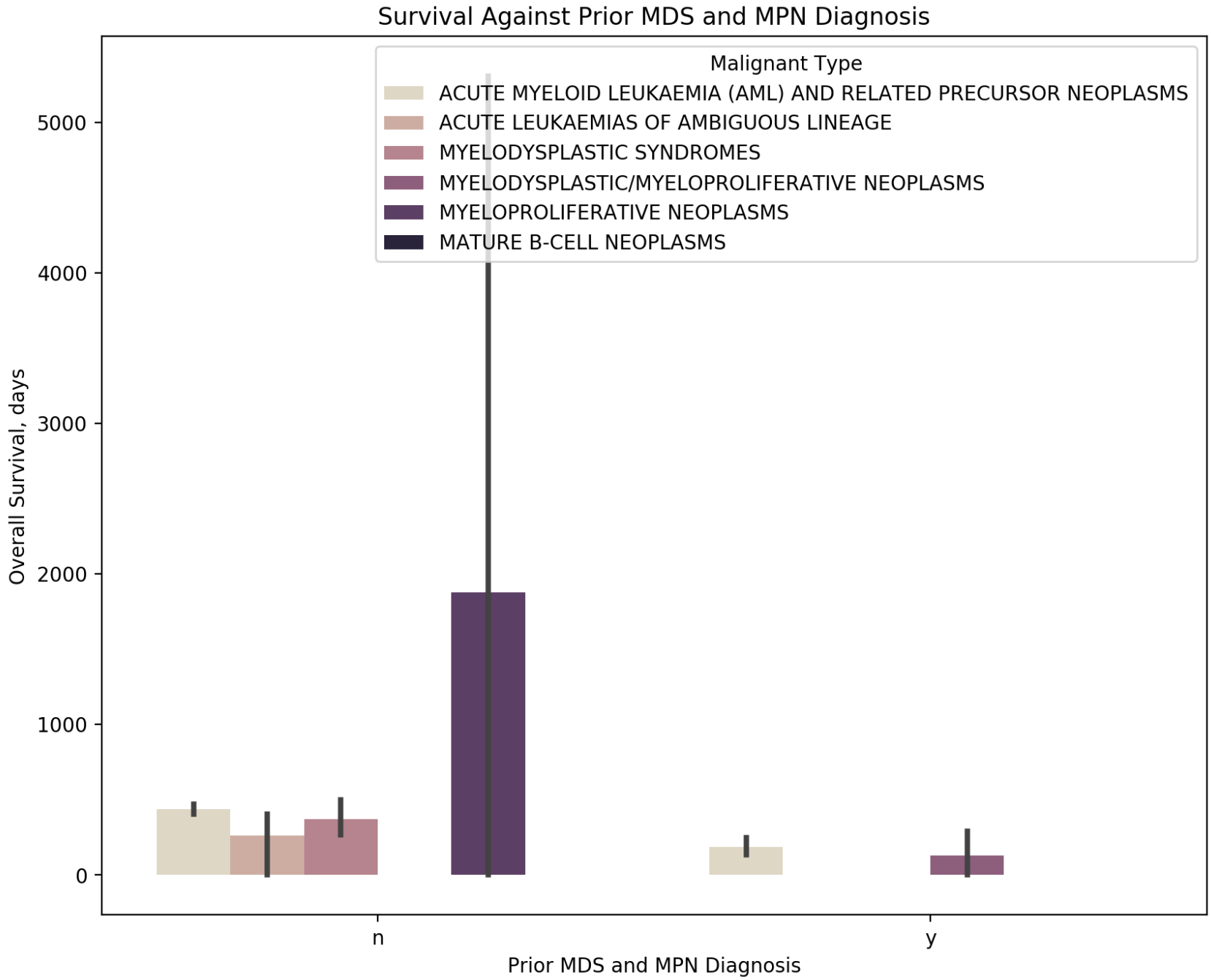
In [86]:
```
sns.barplot(data= clsm_cut,x = 'priorMDS', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('Prior MDS Diagnosis')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against Prior MDS Diagnosis")
```

Out[86]:
```
Text(0.5, 1.0, 'Survival Against Prior MDS Diagnosis')
```



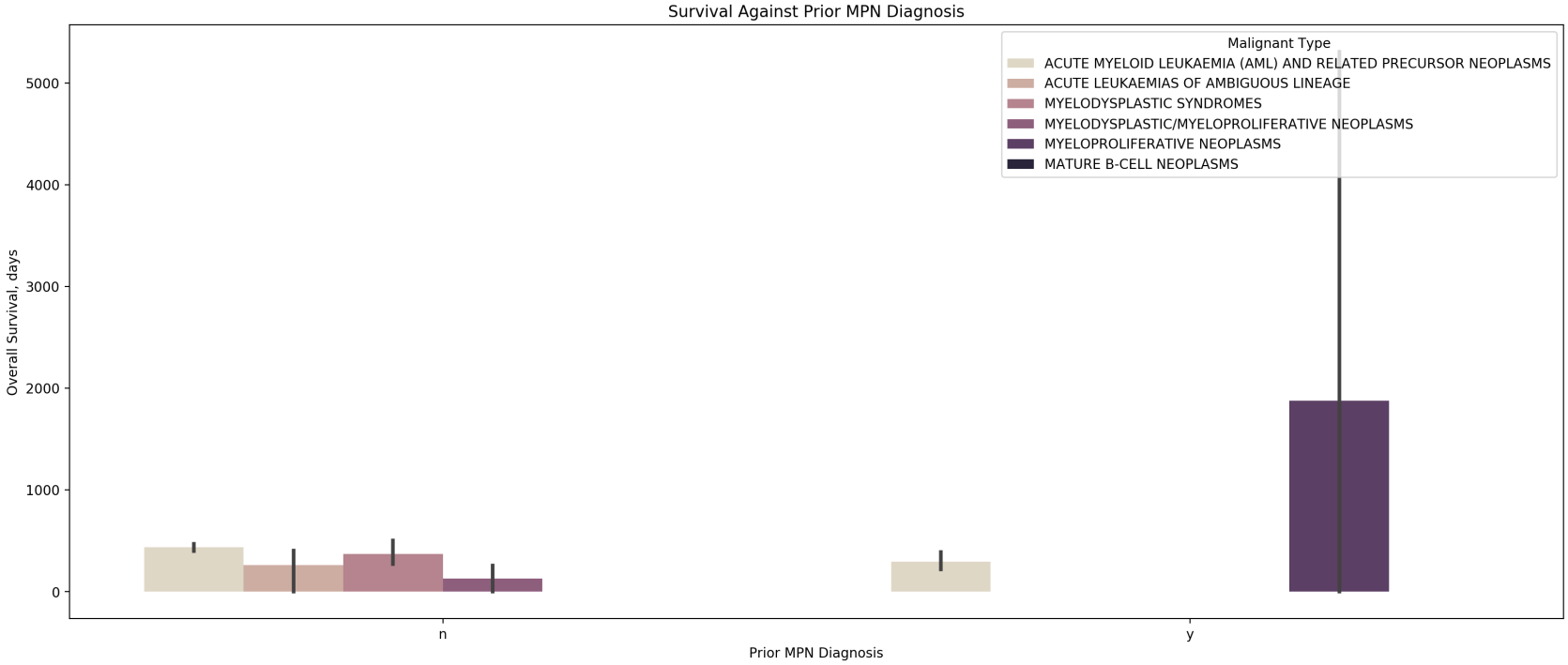
In [87]:
```
sns.barplot(data= clsm_cut,x = 'priorMDSMPN', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(10, 8)
plt.xlabel('Prior MDS and MPN Diagnosis')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against Prior MDS and MPN Diagnosis")
```

Out[87]:
```
Text(0.5, 1.0, 'Survival Against Prior MDS and MPN Diagnosis')
```


Survival Against Prior MDS and MPN Diagnosis
Malignant Type
ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS
ACUTE LEUKAEMIAS OF AMBIGUOUS LINEAGE
MYELODYSPLASTIC SYNDROMES
MYELODYSPLASTIC/MYELOPROLIFERATIVE NEOPLASMS
MYELOPROLIFERATIVE NEOPLASMS
MATURE B-CELL NEOPLASMS
Overall Survival, days
5000
4000
3000
2000
1000
0
n
y
Prior MDS and MPN Diagnosis

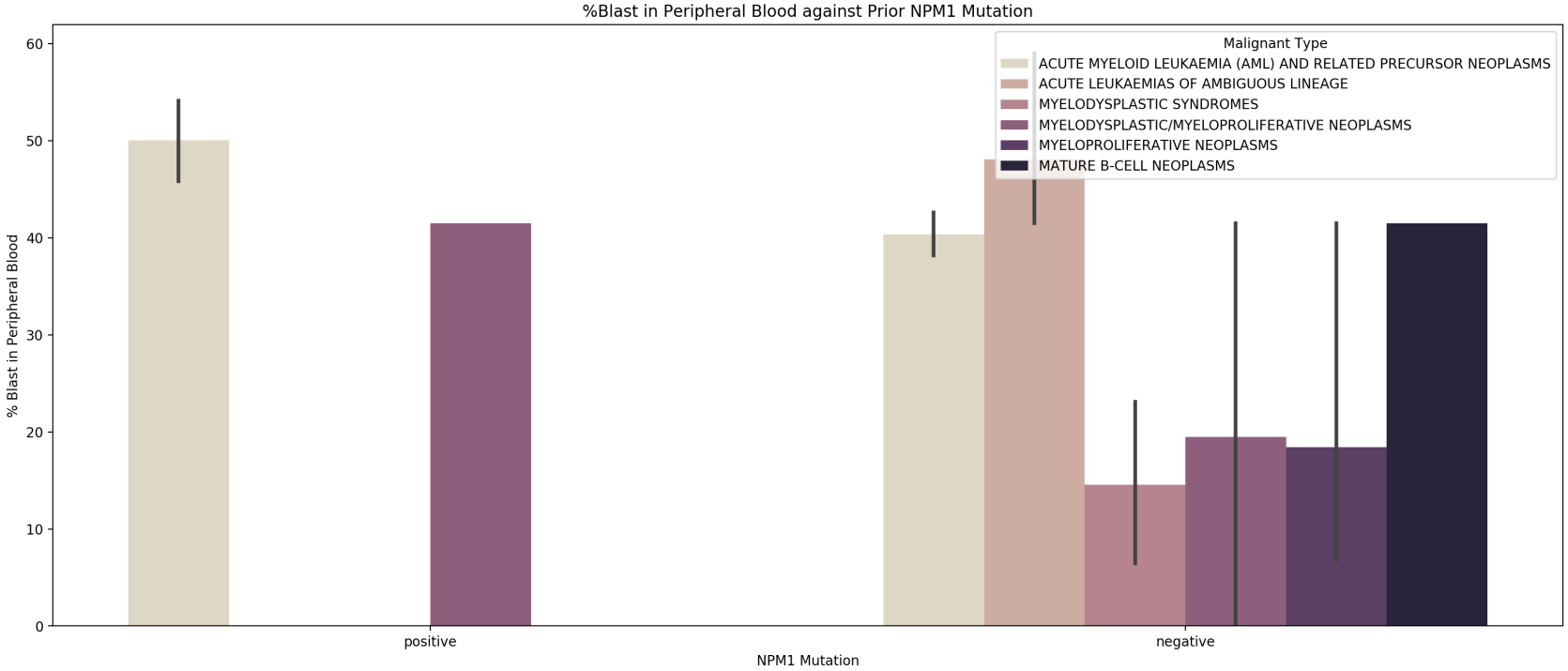
In [88]:
```
sns.barplot(data= clsm_cut,x = 'priorMPN', y = 'overallSurvival',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('Prior MPN Diagnosis')
plt.ylabel('Overall Survival, days')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("Survival Against Prior MPN Diagnosis")
```

Out[88]:
```
Text(0.5, 1.0, 'Survival Against Prior MPN Diagnosis')
```



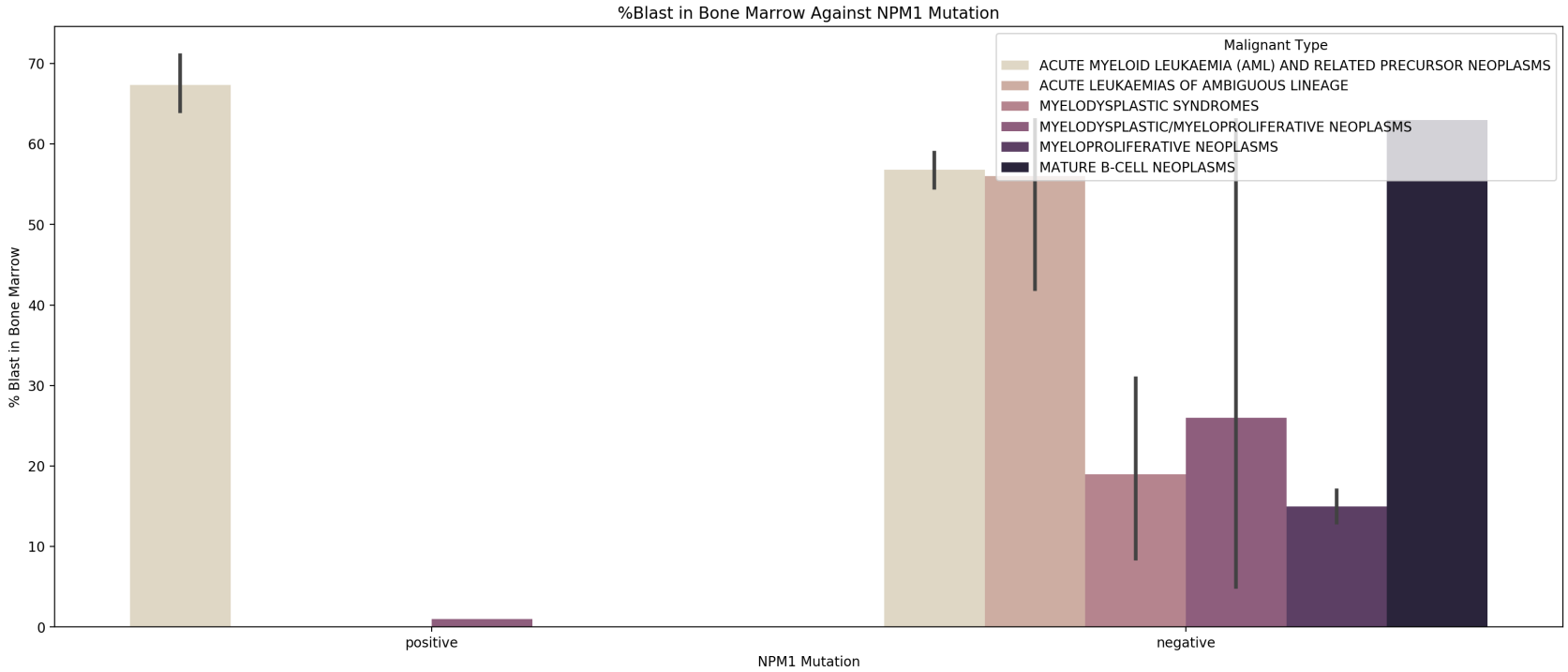
In [89]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = '%.Blasts.in.PB',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Peripheral Blood')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("%Blast in Peripheral Blood against Prior NPM1 Mutation")
```

Out[89]:
```
Text(0.5, 1.0, '%Blast in Peripheral Blood against Prior NPM1 Mutation')
```



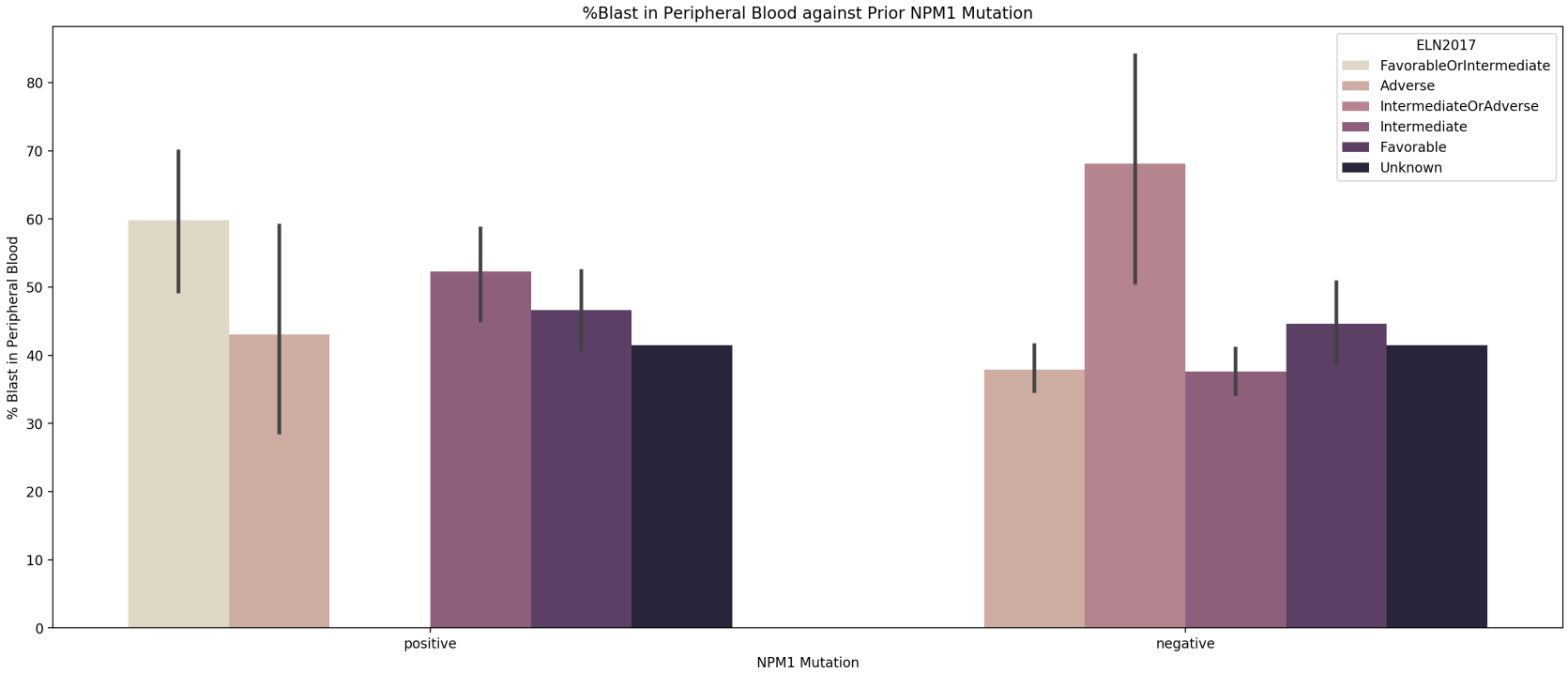
In [90]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = '%.Blasts.in.BM',
            hue = 'dxAtSpecimenAcquisition', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Bone Marrow')
plt.legend(loc='upper right', title = 'Malignant Type')
plt.title("%Blast in Bone Marrow Against NPM1 Mutation")
```

Out[90]:
```
Text(0.5, 1.0, '%Blast in Bone Marrow Against NPM1 Mutation')
```



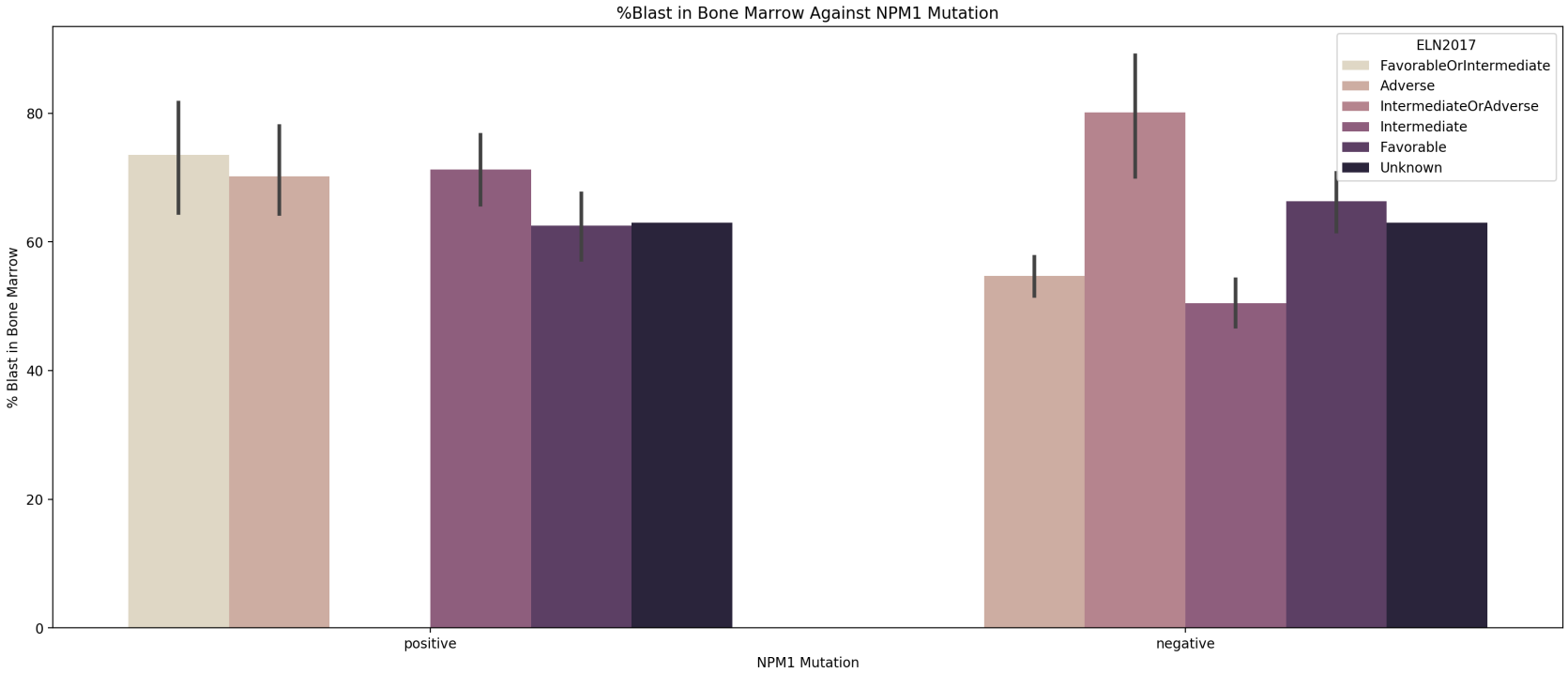
In [91]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = '%.Blasts.in.PB',
            hue = 'ELN2017', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Peripheral Blood')
plt.legend(loc='upper right', title = 'ELN2017')
plt.title("%Blast in Peripheral Blood against Prior NPM1 Mutation")
```

Out[91]:
```
Text(0.5, 1.0, '%Blast in Peripheral Blood against Prior NPM1 Mutation')
```



In [92]:
```
sns.barplot(data= clsm_cut,x = 'NPM1', y = '%.Blasts.in.BM',
            hue = 'ELN2017', palette = "ch:s=-.2,r=.6")
plt.gcf().set_size_inches(20, 8)
plt.xlabel('NPM1 Mutation')
plt.ylabel('% Blast in Bone Marrow')
plt.legend(loc='upper right', title = 'ELN2017')
plt.title("%Blast in Bone Marrow Against NPM1 Mutation")
```

Out[92]:
```
Text(0.5, 1.0, '%Blast in Bone Marrow Against NPM1 Mutation')
```



Transformation: Final Prep prior to Data Modeling

Create Target Variable

In [93]:
```
clsm_cut['dxAtSpecimenAcquisition'].value_counts()
```

Out[93]:
```
ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS    646
MYELODYSPLASTIC SYNDROMES                                         15
MYELODYSPLASTIC/MYELOPROLIFERATIVE NEOPLASMS                       4
ACUTE LEUKAEMIAS OF AMBIGUOUS LINEAGE                              3
MYELOPROLIFERATIVE NEOPLASMS                                       3
MATURE B-CELL NEOPLASMS                                            1
Name: dxAtSpecimenAcquisition, dtype: int64
```

In [94]:
```
#create column for AML detected
clsm_cut['AML_detected'] = ['yes' if x == 'ACUTE MYELOID LEUKAEMIA (AML) AND RELATED PRECURSOR NEOPLASMS'
                            else 'no' for x in clsm_cut['dxAtSpecimenAcquisition']]
clsm_cut.head()
```

Out[94]:

	LabId	PatientId	consensus_sex	inferred_ethnicity	isRelapse	isTransformed	priorMalignancyNonMyeloid	priorMDS	priorMDSMPN
0	09-00705	163	Male	White	False	False	n	n	n
1	10-00136	174	Male	White	False	False	n	n	n
2	10-00172	175	Female	White	False	False	n	n	n
3	10-00507	45	Female	White	False	False	n	n	n
4	10-00542	174	Male	White	True	False	n	n	n

In [95]:
```
clsm_cut['AML_detected'].value_counts()
```

Out[95]:
```
yes    646
no      26
Name: AML_detected, dtype: int64
```

New Dataframe for SageMaker JumpStart Regression Model

Transform select categorical attributes to numerical:

In [96]:
```
#AML_detected
clsm_cut['AML_detected'].replace(['no', 'yes'],
                         [0, 1], inplace=True)
#npm1
clsm_cut['NPM1'].replace(['negative', 'positive'],
                         [0, 1], inplace=True)
#flt3-itd
clsm_cut['FLT3-ITD'].replace(['negative', 'positive'],
                         [0, 1], inplace=True)
#priormalignancynonmyeloid
clsm_cut['priorMalignancyNonMyeloid'].replace(['n', 'y'],
                         [0, 1], inplace=True)
#priormds
clsm_cut['priorMDS'].replace(['n', 'y'],
                         [0, 1], inplace=True)
#priormdsmpn
clsm_cut['priorMDSMPN'].replace(['n', 'y'],
                         [0, 1], inplace=True)
#priormpn
clsm_cut['priorMPN'].replace(['n', 'y'],
                         [0, 1], inplace=True)
```

In [97]:
```
#Create new dataframe with transformed attributes and necessary attributes
clsm_cut_transform = pd.DataFrame(clsm_cut[['AML_detected', 'NPM1', 'FLT3-ITD', 'isRelapse', 'isTransformed',
                                'priorMalignancyNonMyeloid', 'priorMDS', 'priorMDSMPN', 'priorMPN',
                                '%.Blasts.in.PB', '%.Blasts.in.BM', 'overallSurvival']])
```

In [98]:
```
#Transform data type:
clsm_cut_transform['NPM1'] = clsm_cut['NPM1'].astype(int)
clsm_cut_transform['FLT3-ITD'] = clsm_cut['FLT3-ITD'].astype(int)
clsm_cut_transform['isRelapse'] = clsm_cut['isRelapse'].astype(int)
clsm_cut_transform['isTransformed'] = clsm_cut['isTransformed'].astype(int)
```

New Numerical Dataframe Correlation Matrix

In [99]:
```
clsm_cut_transform.corr()
```

Out[99]:

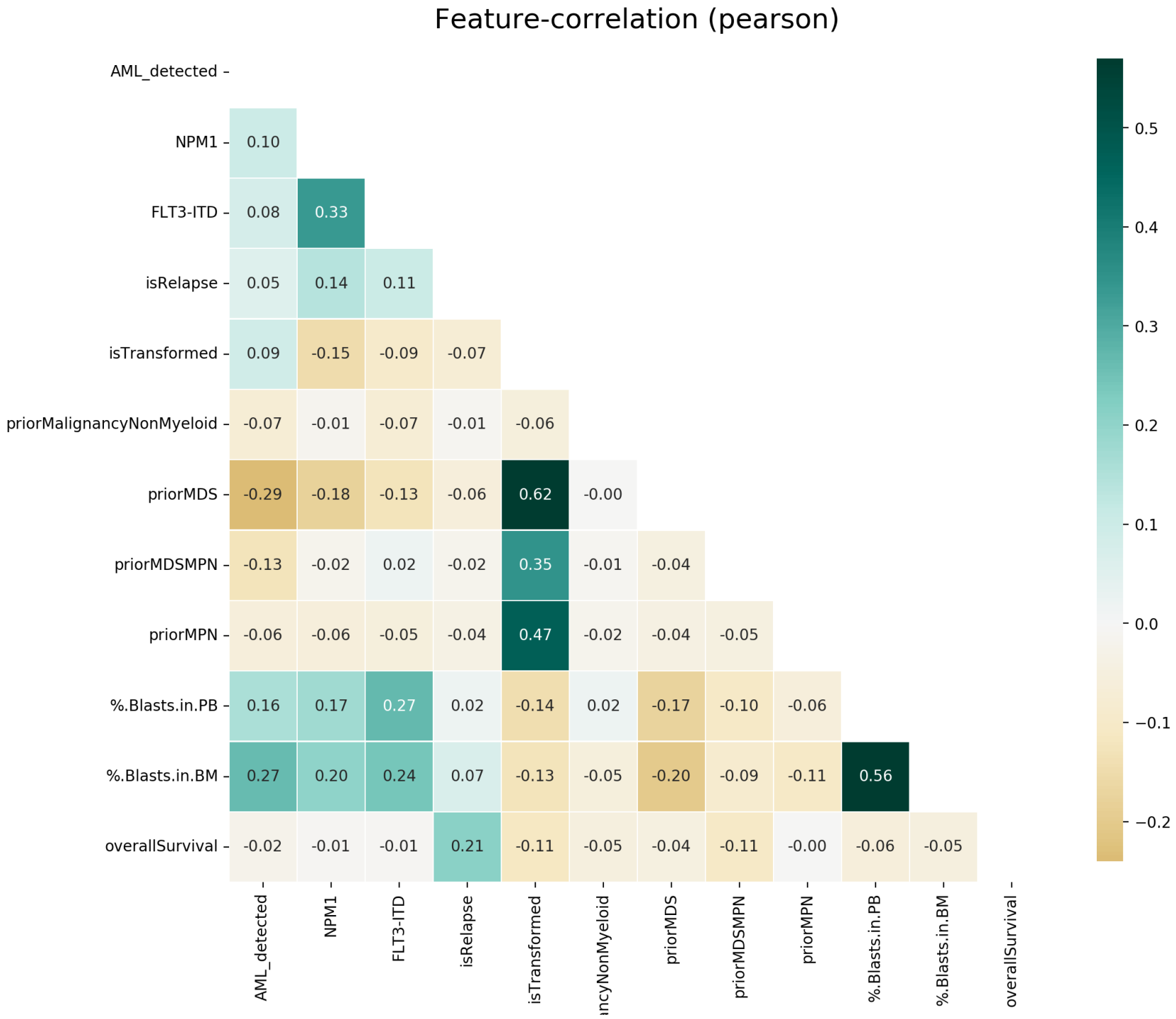
	AML_detected	NPM1	FLT3-ITD	isRelapse	isTransformed	priorMalignancyNonMyeloid	priorMDS	pri
AML_detected	1.000000	0.098997	0.077525	0.054383	0.089238	-7.245182e-02	-2.912038e-01	
NPM1	0.098997	1.000000	0.333543	0.140481	-0.148233	-1.257739e-02	-1.771024e-01	
FLT3-ITD	0.077525	0.333543	1.000000	0.107818	-0.092782	-7.228395e-02	-1.272762e-01	
isRelapse	0.054383	0.140481	0.107818	1.000000	-0.072971	-9.623173e-03	-6.051426e-02	
isTransformed	0.089238	-0.148233	-0.092782	-0.072971	1.000000	-5.562376e-02	6.200179e-01	
priorMalignancyNonMyeloid	-0.072452	-0.012577	-0.072284	-0.009623	-0.055624	1.000000e+00	-1.056121e-17	
priorMDS	-0.291204	-0.177102	-0.127276	-0.060514	0.620018	-1.056121e-17	1.000000e+00	
priorMDSMPN	-0.127707	-0.019763	0.022049	-0.020414	0.346275	-9.820928e-03	-4.405654e-02	
priorMPN	-0.057154	-0.059377	-0.054524	-0.037020	0.472862	-1.913898e-02	-4.227151e-02	
%.Blasts.in.PB	0.155752	0.174675	0.271851	0.020293	-0.141930	2.215108e-02	-1.739113e-01	
%.Blasts.in.BM	0.265724	0.201114	0.242420	0.069284	-0.128224	-5.402601e-02	-2.015171e-01	
overallSurvival	-0.022216	-0.006728	-0.008150	0.210147	-0.113017	-4.893419e-02	-4.466673e-02	

In [100…

```
klib.corr_plot(clsm_cut_transform)
```

Out[100]:

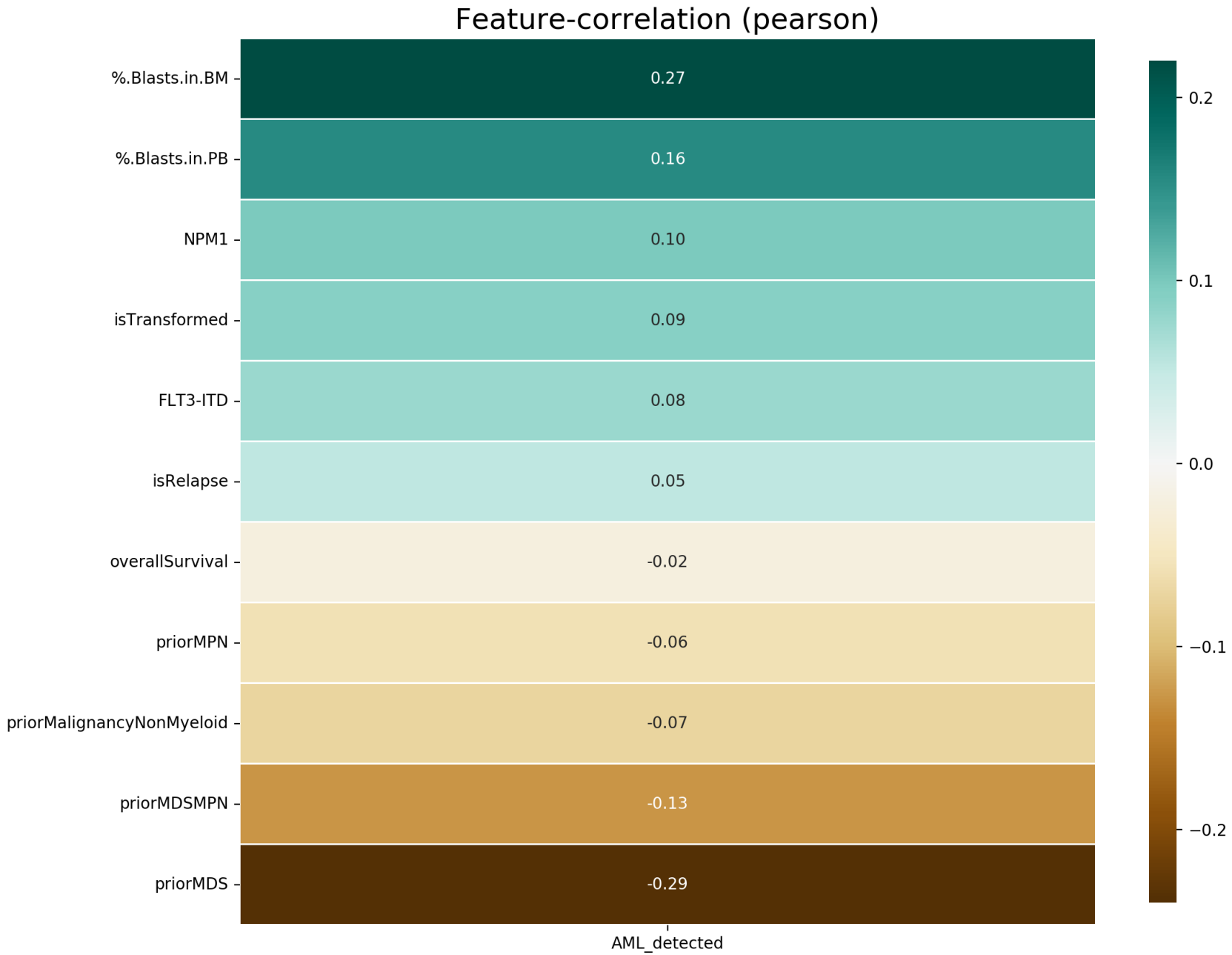
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8a1237a50>
```


Feature-correlation (pearson)
AML_detected
NPM1
FLT3-ITD
isRelapse
isTransformed
priorMalignancyNonMyeloid
priorMDS
priorMDSMPN
priorMPN
%.Blasts.in.PB
%.Blasts.in.BM
overallSurvival
0.10
0.08 0.33
0.05 0.14 0.11
0.09 -0.15 -0.09 -0.07
-0.07 -0.01 -0.07 -0.01 -0.06
-0.29 -0.18 -0.13 -0.06 0.62 -0.00
-0.13 -0.02 0.02 -0.02 0.35 -0.01 -0.04
-0.06 -0.06 -0.05 -0.04 0.47 -0.02 -0.04 -0.05
0.16 0.17 0.27 0.02 -0.14 0.02 -0.17 -0.10 -0.06
0.27 0.20 0.24 0.07 -0.13 -0.05 -0.20 -0.09 -0.11 0.56
-0.02 -0.01 -0.01 0.21 -0.11 -0.05 -0.04 -0.11 -0.00 -0.06 -0.05
AML_detected
NPM1
FLT3-ITD
isRelapse
isTransformed
incyNonMyeloid
priorMDS
priorMDSMPN
priorMPN
%.Blasts.in.PB
%.Blasts.in.BM
overallSurvival
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2

priorMalign

In [101…
```
klib.corr_plot(clsm_cut_transform, target='AML_detected')
```

Out[101]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb8a110e550>
```



One-Hot Encoding

In [102…
```
clsm_t = pd.get_dummies(clsm_cut_transform, columns= ['NPM1', 'FLT3-ITD', 'priorMalignancyNonMyeloid',
                                                      'priorMDS', 'priorMDSMPN', 'priorMPN', 'isRelapse',
                                                      'isTransformed'])
```

In [103…
```
clsm_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   AML_detected                 672 non-null    int64
 1   %.Blasts.in.PB               672 non-null    float64
 2   %.Blasts.in.BM               672 non-null    float64
 3   overallSurvival              672 non-null    float64
 4   NPM1_0                       672 non-null    uint8
 5   NPM1_1                       672 non-null    uint8
 6   FLT3-ITD_0                   672 non-null    uint8
 7   FLT3-ITD_1                   672 non-null    uint8
 8   priorMalignancyNonMyeloid_0  672 non-null    uint8
 9   priorMalignancyNonMyeloid_1  672 non-null    uint8
 10  priorMDS_0                   672 non-null    uint8
 11  priorMDS_1                   672 non-null    uint8
 12  priorMDSMPN_0                672 non-null    uint8
 13  priorMDSMPN_1                672 non-null    uint8
 14  priorMPN_0                   672 non-null    uint8
 15  priorMPN_1                   672 non-null    uint8
 16  isRelapse_0                  672 non-null    uint8
 17  isRelapse_1                  672 non-null    uint8
 18  isTransformed_0              672 non-null    uint8
 19  isTransformed_1              672 non-null    uint8
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

In [104…
```
clsm_t.head()
```

Out[104]:

	AML_detected	%.Blasts.in.PB	%.Blasts.in.BM	overallSurvival	NPM1_0	NPM1_1	FLT3-ITD_0	FLT3-ITD_1	priorMalignancyNonMyeloid_0	priorMal
0	1	97.0	94.0	425.0	0	1	0	1	1	
1	1	19.0	80.0	419.0	1	0	0	1	1	
2	1	99.0	91.0	541.0	1	0	0	1	1	
3	1	97.0	97.0	511.0	0	1	0	1	1	
4	1	80.0	87.0	419.0	1	0	0	1	1	

In [105…

```
#Transform headers
clsm_t = clsm_t.rename(columns={ '%.Blasts.in.PB': 'Feature_1', '%.Blasts.in.BM': 'Feature_2',
                                 'overallSurvival': 'Feature_3',
                                  'NPM1_0': 'Feature_4', 'NPM1_1': 'Feature_5',
                                 'FLT3-ITD_0': 'Feature_6', 'FLT3-ITD_1': 'Feature_7',
                                  'priorMalignancyNonMyeloid_0': 'Feature_8', 'priorMalignancyNonMyeloid_1'
                                  'priorMDS_0': 'Feature_10', 'priorMDS_1': 'Feature_11',
                                 'priorMDSMPN_0': 'Feature_12', 'priorMDSMPN_1': 'Feature_13',
                                  'priorMPN_0': 'Feature_14', 'priorMPN_1': 'Feature_15',
                                  'isRelapse_0': 'Feature_16', 'isRelapse_1': 'Feature_17',
                                 'isTransformed_0': 'Feature_18', 'isTransformed_1': 'Feature_19' })
```

In [106…

```
clsm_t.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 672 entries, 0 to 671
Data columns (total 20 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   AML_detected  672 non-null    int64
 1   Feature_1     672 non-null    float64
 2   Feature_2     672 non-null    float64
 3   Feature_3     672 non-null    float64
 4   Feature_4     672 non-null    uint8
 5   Feature_5     672 non-null    uint8
 6   Feature_6     672 non-null    uint8
 7   Feature_7     672 non-null    uint8
 8   Feature_8     672 non-null    uint8
 9   Feature_9     672 non-null    uint8
 10  Feature_10    672 non-null    uint8
 11  Feature_11    672 non-null    uint8
 12  Feature_12    672 non-null    uint8
 13  Feature_13    672 non-null    uint8
 14  Feature_14    672 non-null    uint8
 15  Feature_15    672 non-null    uint8
 16  Feature_16    672 non-null    uint8
 17  Feature_17    672 non-null    uint8
 18  Feature_18    672 non-null    uint8
 19  Feature_19    672 non-null    uint8
dtypes: float64(3), int64(1), uint8(16)
memory usage: 31.6 KB
```

Split the Data into Train, Test, and Validation Sets

In [129…

```
clsm_t
```

Out[129]:

	AML_detected	Feature_1	Feature_2	Feature_3	Feature_4	Feature_5	Feature_6	Feature_7	Feature_8	Feature_9	Feature_10	Featur
0	1	97.0	94.0	425.0	0	1	0	1	1	0	1	
1	1	19.0	80.0	419.0	1	0	0	1	1	0	1	
2	1	99.0	91.0	541.0	1	0	0	1	1	0	1	
3	1	97.0	97.0	511.0	0	1	0	1	1	0	1	
4	1	80.0	87.0	419.0	1	0	0	1	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	
667	1	53.2	63.0	362.0	1	0	1	0	1	0	1	
668	1	74.0	90.0	323.0	1	0	1	0	1	0	1	
669	1	48.0	63.0	323.0	1	0	0	1	1	0	1	
670	1	41.5	20.0	153.0	1	0	1	0	1	0	1	
671	1	41.5	63.0	256.0	1	0	1	0	1	0	1	

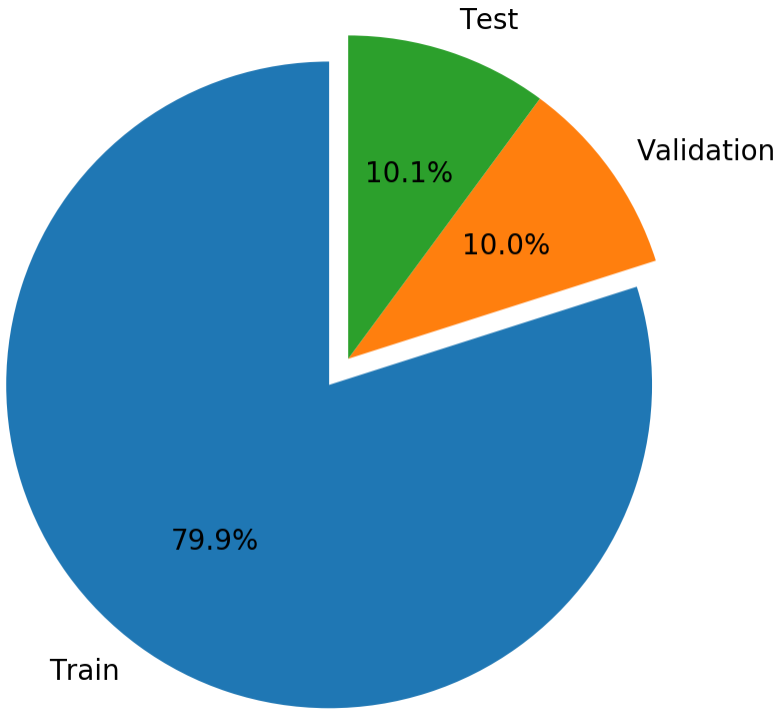
672 rows × 20 columns

In [130…

```
from sklearn.model_selection import train_test_split

# Split all data into 80% train and 20% holdout
clsm_train, clsm_holdout = train_test_split(clsm_t, test_size=0.20, random_state=42)

# Split holdout data into 50% validation and 50% test
clsm_validation, clsm_test = train_test_split(clsm_holdout, test_size=0.50, random_state=42)
```

In [131…

```
# Pie chart, where the slices will be ordered and plotted counter-clockwise:
labels = ["Train", "Validation", "Test"]
sizes = [len(clsm_train.index), len(clsm_validation.index), len(clsm_test.index)]
explode = (0.1, 0, 0)

fig1, ax1 = plt.subplots()

ax1.pie(sizes, explode=explode, labels=labels, autopct="%1.1f%%", startangle=90)

# Equal aspect ratio ensures that pie is drawn as a circle.
ax1.axis("equal")

plt.show()
```



In [132…

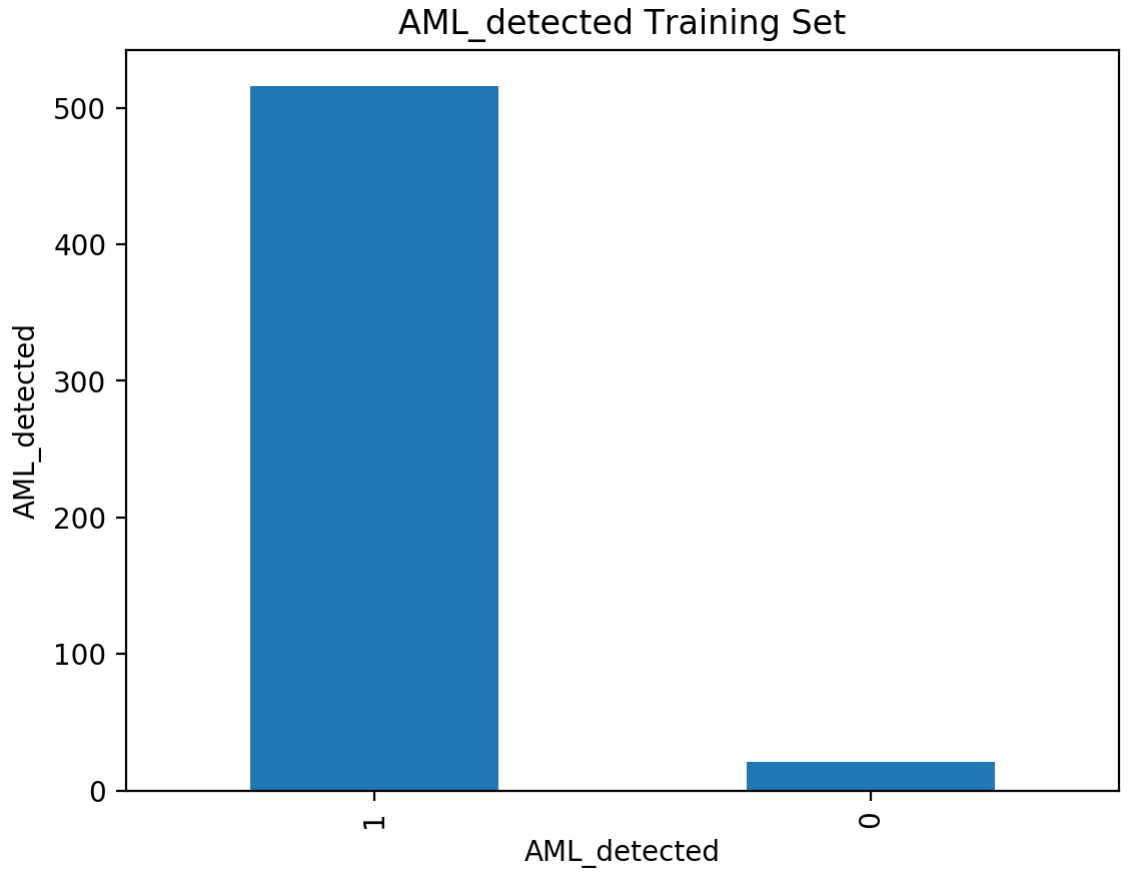
```
#Show 80% Train Data Split
clsm_train.shape
```

Out[132]:

```
(537, 20)
```

In [133…

```
clsm_train["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Training Set")
plt.xlabel("AML_detected")
plt.ylabel("AML_detected")

plt.show()
```



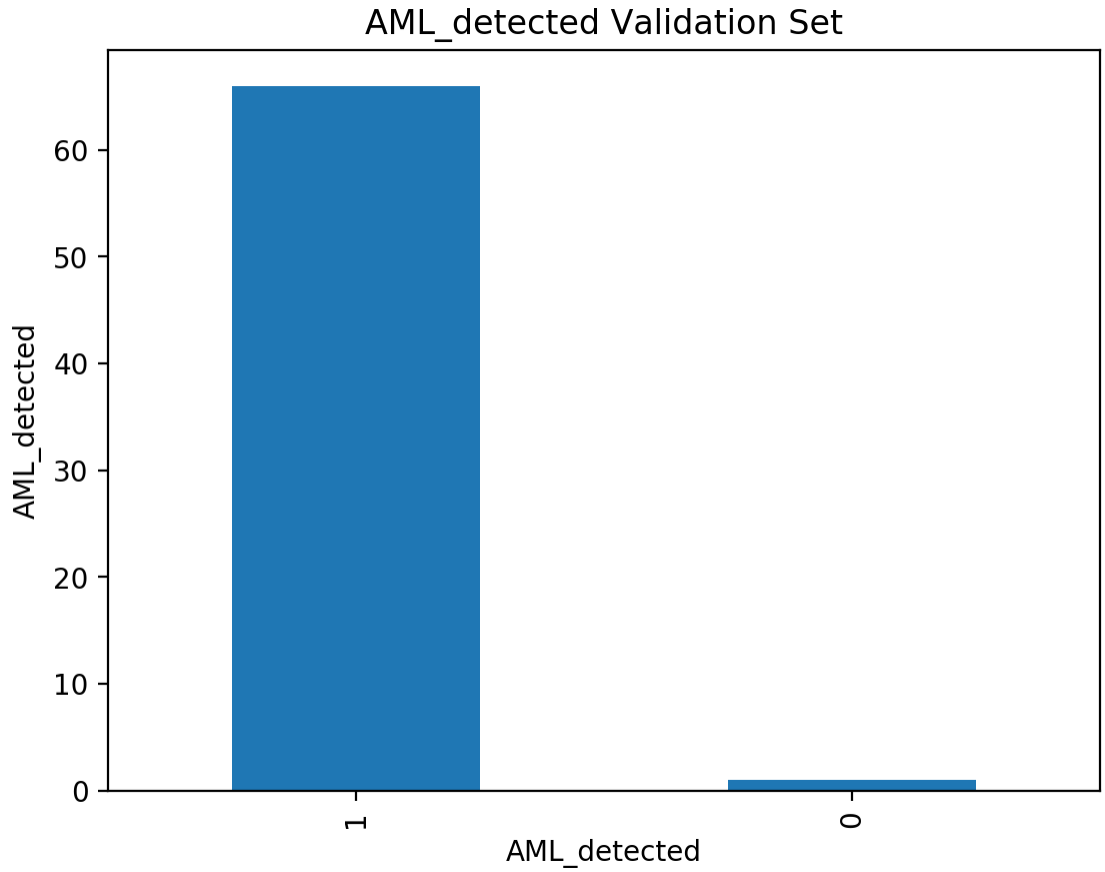
In [134…

```
#Show 5% Validation Data Split
clsm_validation.shape
```

Out[134]:

```
(67, 20)
```

In [135…
```
clsm_validation["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Validation Set")
plt.xlabel("AML_detected")
plt.ylabel("AML_detected")

plt.show()
```

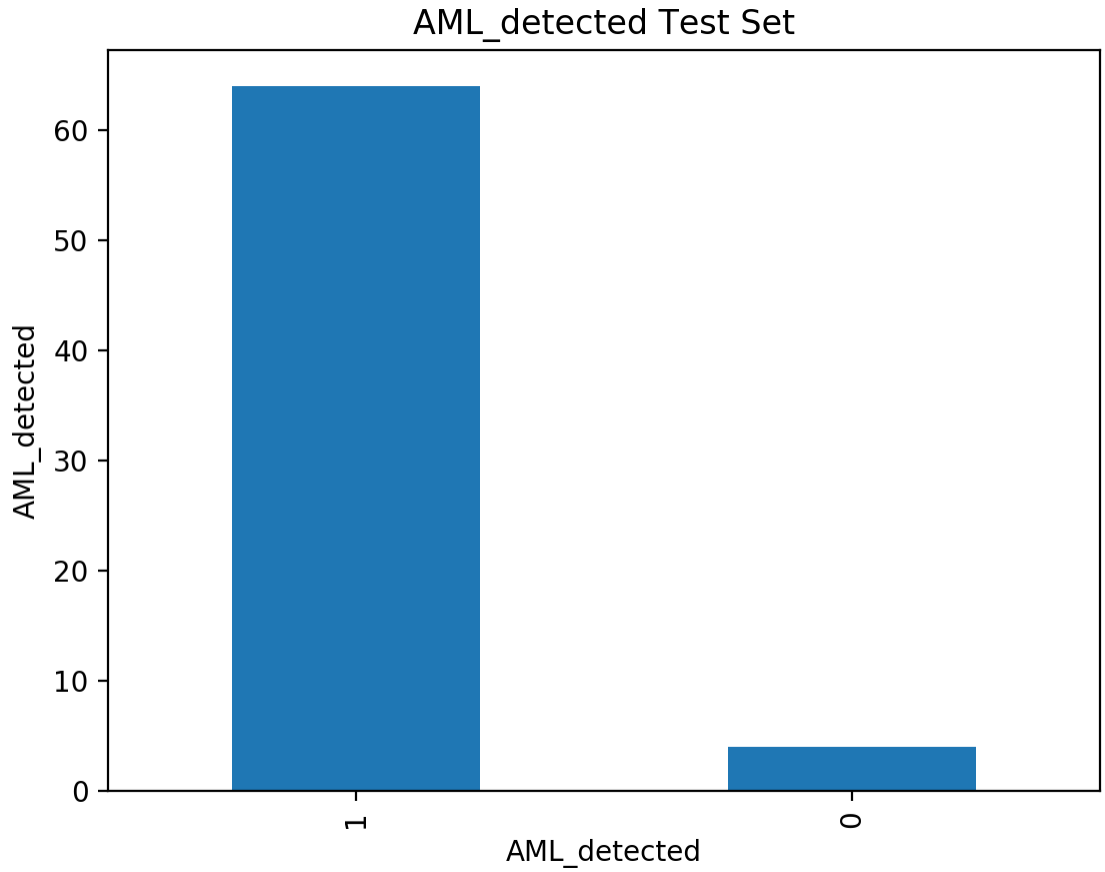


In [138…
```
#Show 5% Test Data Split
clsm_test.shape
```

Out[138]:
```
(68, 20)
```

In [139…

```
clsm_test["AML_detected"].value_counts().plot(kind="bar", title="AML_detected Test Set")
plt.xlabel("AML_detected")
plt.ylabel("AML_detected")

plt.show()
```



Balance Training Data

In [140…

```
clsm_train['AML_detected'].value_counts()
```

Out[140]:

```
1    516
0     21
Name: AML_detected, dtype: int64
```

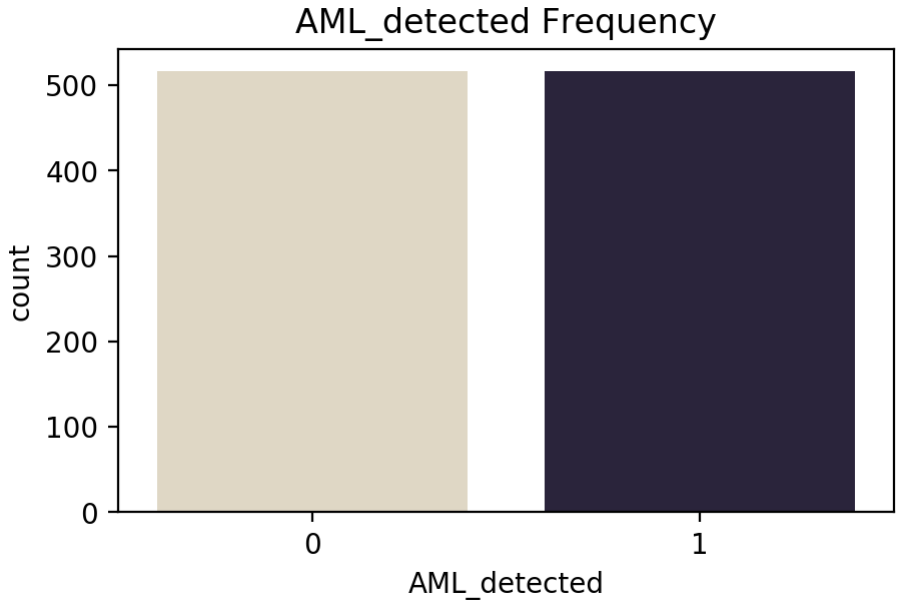
In [141…
```
#resampling of training data set
to_resample= clsm_train.loc[clsm_train['AML_detected'] == 0] #isolate all records of AML_detected
our_resample=to_resample.sample(n=495, replace=True) #sample w/ replacement
clsm_t_rebal=pd.concat([clsm_train, our_resample]) #combine original training set w/ resampled records
clsm_t_rebal['AML_detected'].value_counts()
```

Out[141]:
```
1    516
0    516
Name: AML_detected, dtype: int64
```

In [142…
```
clsm_t_rebal.shape
```

Out[142]:
```
(1032, 20)
```

In [143…
```
sns.countplot(x=clsm_t_rebal["AML_detected"], palette = "ch:s=-.2,r=.6")
plt.xlabel('AML_detected')
plt.title('AML_detected Frequency')
plt.gcf().set_size_inches(5, 3)
```



Model: Logistic Regression

In [145…

```
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import NearestNeighbors, KNeighborsClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score, plot_confusion_matrix, classification_report, confusion_matrix, roc_curve
import statistics as stats
```

In [146…

```
#Define/List the features
X_var = list(clsm_t.columns)
X_var
```

Out[146]:

```
['AML_detected',
 'Feature_1',
 'Feature_2',
 'Feature_3',
 'Feature_4',
 'Feature_5',
 'Feature_6',
 'Feature_7',
 'Feature_8',
 'Feature_9',
 'Feature_10',
 'Feature_11',
 'Feature_12',
 'Feature_13',
 'Feature_14',
 'Feature_15',
 'Feature_16',
 'Feature_17',
 'Feature_18',
 'Feature_19']
```

In [147…

```
#Define the target
target ='AML_detected'
X_var.remove(target)

x_train = clsm_t_rebal[X_var]
y_train = clsm_t_rebal[target]
x_test = clsm_test[X_var]
y_test = clsm_test[target]
x_valid = clsm_validation[X_var]
y_valid = clsm_validation[target]
```

In [148…
```
x_train.shape
```

Out[148]:
```
(1032, 19)
```

In [149…
```
x_valid.shape
```

Out[149]:
```
(67, 19)
```

In [150…
```
x_test.shape
```

Out[150]:
```
(68, 19)
```

In [151…
```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.fit_transform(x_test)
```

In [152…
```
x_train_scaled
```

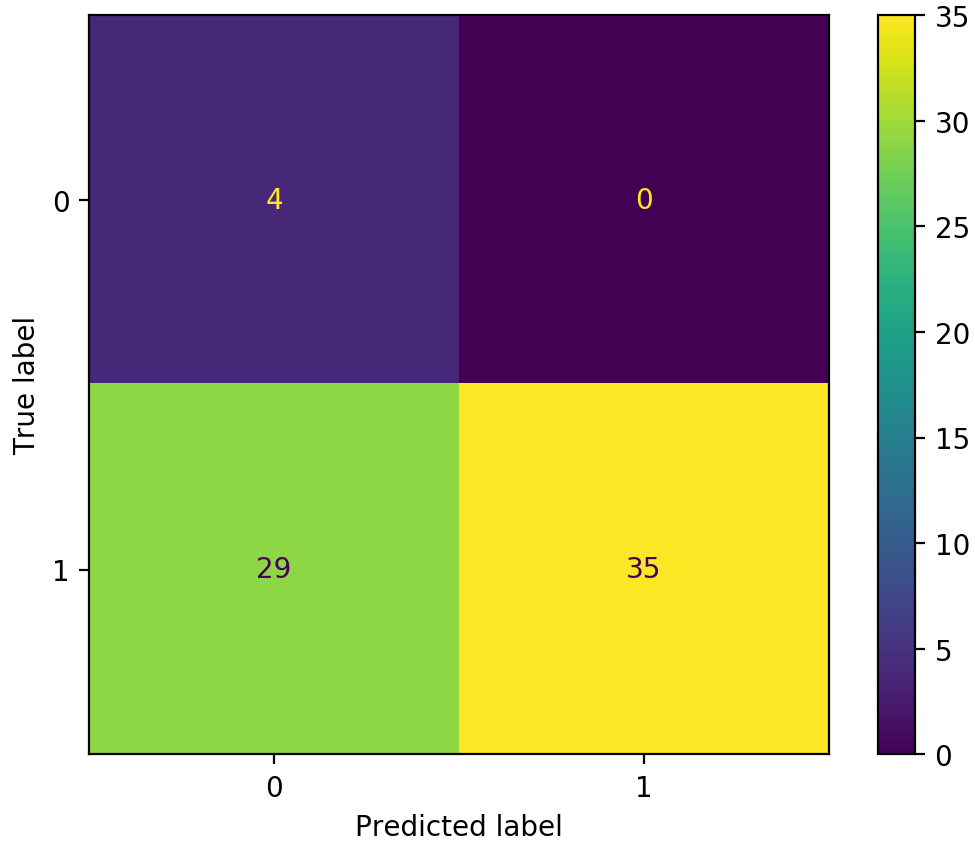
Out[152]:
```
array([[ 2.46104494,  1.73896145, -0.31081992, ..., -0.18174992,
         0.31841389, -0.31841389],
       [ 0.98032538, -0.14328915, -0.33863929, ..., -0.18174992,
         0.31841389, -0.31841389],
       [ 0.37634767,  0.64383383,  1.955194  , ..., -0.18174992,
         0.31841389, -0.31841389],
       ...,
       [ 0.37634767, -1.47797593, -0.32725864, ..., -0.18174992,
         0.31841389, -0.31841389],
       [-0.96798983, -0.93041212, -0.15654888, ..., -0.18174992,
         0.31841389, -0.31841389],
       [-1.16282135, -1.23841677, -0.15654888, ..., -0.18174992,
         0.31841389, -0.31841389]])
```

In [153…
```
x_test_scaled
```

Out[153]:
```
array([[ 0.53249353, -0.20540826, -1.10042836, ..., -0.28171808,
         0.43929769, -0.43929769],
       [-0.75236183, -0.77474636,  0.98176022, ..., -0.28171808,
         0.43929769, -0.43929769],
       [-0.56936728, -1.19226097,  0.43006922, ..., -0.28171808,
         0.43929769, -0.43929769],
       ...,
       [ 0.00687088,  0.25006223,  0.24956203, ..., -0.28171808,
         0.43929769, -0.43929769],
       [-0.90810187,  0.59166509,  2.57327432, ..., -0.28171808,
         0.43929769, -0.43929769],
       [ 0.04191239, -1.0024816 , -0.67839747, ..., -0.28171808,
         0.43929769, -0.43929769]])
```

In [154…
```
#Logistic Regression Model
logit_reg = LogisticRegression(random_state=27)
logit_reg.fit(x_train_scaled, y_train)
y_pred = logit_reg.predict(x_test)

#Predict on validation set
logit_reg_pred1 = logit_reg.predict(x_valid)

plot_confusion_matrix(logit_reg, x_test_scaled, y_test)
plt.grid(False)

#accuracy and classification report on untuned model
print('Untuned Logistic Regression Model')
print('Accuracy Score')
print(accuracy_score(y_valid, logit_reg_pred1))
print('Cross Validation: \n',
      classification_report(y_valid, logit_reg_pred1))
```

```
Untuned Logistic Regression Model
Accuracy Score
0.9850746268656716
Cross Validation:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.99      1.00      0.99        66

    accuracy                           0.99        67
   macro avg       0.49      0.50      0.50        67
weighted avg       0.97      0.99      0.98        67
```



In [155…

```
#Tune logistic regression model using RandomizedSearchCV() and cross validate with repeated stratified kfold with fiv
#Generate overall best accuracy score with optimal hyperparameters
from sklearn.model_selection import RepeatedStratifiedKFold, RandomizedSearchCV
from scipy.stats import loguniform
model1 = LogisticRegression(random_state=27)
cv = RepeatedStratifiedKFold(n_splits=5, n_repeats=2, random_state=27)

space = dict()

# define search space
space['solver'] = ['newton-cg', 'lbfgs', 'liblinear']
space['penalty'] = ['none', 'l1', 'l2', 'elasticnet']
space['C'] = loguniform(1e-5, 100)

# define search
search = RandomizedSearchCV(model1, space,
                            scoring='accuracy',
n_jobs=-1, cv=cv, random_state=777)

# execute search
result = search.fit(x_train, y_train)

# summarize result
print('Best Score: %s' % result.best_score_)
print('Best Hyperparameters: %s' % result.best_params_)
```

```
Best Score: 0.9103489517377235
Best Hyperparameters: {'C': 0.005198849908368508, 'penalty': 'none', 'solver': 'lbfgs'}
```

Release Resources

In [96]:
```
%%html

<p><b>Shutting down your kernel for this notebook to release resources.</b></p>
<button class="sm-command-button" data-commandlinker-command="kernelmenu:shutdown" style="display:none;">Shutdown Ker

<script>
try {
    els = document.getElementsByClassName("sm-command-button");
    els[0].click();
}
catch(err) {
    // NoOp
}
</script>
```

Shutting down your kernel for this notebook to release resources.

In [132…
```
%%javascript

try {
    Jupyter.notebook.save_checkpoint();
    Jupyter.notebook.session.delete();
}
catch(err) {
    // NoOp
}
```