Testicular Cancer Survival Prediction using Machine Learning

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px # is a high-level interface for creating various types of interactive plots w
import plotly.graph_objects as go # is a lower-level interface that offers more control and customization
```

In [2]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [3]:

```
df = pd.read_csv("Testicular Cancer.csv")
```

In [4]:

```
df.head()
```

Out[4]:

	Study ID	Patient ID	Sample ID	Diagnosis Age	Neoplasm Disease Stage American Joint Committee on Cancer Code	Neoplasm American Joint Committee on Cancer Clinical Distant Metastasis M Stage	American Joint Committee on Cancer Metastasis Stage Code	American Joint Committee on Cancer Lymph Node Stage Code.1	American Joint Committee on Cancer Lymph Node Stage Code	Neoplasm Disease Lymph Node Stage American Joint Committee on Cancer Code	.
0	tgct_tcga	TCGA-2G-AAEW	TCGA-2G-AAEW-01	31.0	Stage IS	M0	M0	N0	T1	N0	.
1	tgct_tcga	TCGA-2G-AAEX	TCGA-2G-AAEX-01	38.0	Stage IS	M0	M0	N0	T1	N0	.
2	tgct_tcga	TCGA-2G-AAF1	TCGA-2G-AAF1-01	28.0	Stage IS	M0	M0	N0	T1	N0	.
3	tgct_tcga	TCGA-2G-AAF4	TCGA-2G-AAF4-01	30.0	Stage IA	M0	M0	N0	T1	N0	.
4	tgct_tcga	TCGA-2G-AAF6	TCGA-2G-AAF6-01	28.0	Stage IS	M0	M0	N0	T1	N0	.

5 rows × 88 columns

In [5]:

```
df.tail()
```

Out[5]:

	Study ID	Patient ID	Sample ID	Diagnosis Age	Neoplasm Disease Stage American Joint Committee on Cancer Code	Neoplasm American Joint Committee on Cancer Clinical Distant Metastasis M Stage	American Joint Committee on Cancer Metastasis Stage Code	American Joint Committee on Cancer Lymph Node Stage Code.1	American Joint Committee on Cancer Lymph Node Stage Code	Neoplasm Disease Lymph Node Stage American Joint Committee on Cancer Code
151	tgct_tcga	TCGA-ZM-AA0D	TCGA-ZM-AA0D-01	34.0	Stage IA	M0	M0	N0	T1	NX
152	tgct_tcga	TCGA-ZM-AA0E	TCGA-ZM-AA0E-01	39.0	Stage IA	M0	M0	N0	T1	NX
153	tgct_tcga	TCGA-ZM-AA0F	TCGA-ZM-AA0F-01	35.0	Stage IA	M0	M0	N0	T1	NX
154	tgct_tcga	TCGA-ZM-AA0H	TCGA-ZM-AA0H-01	50.0	Stage IS	M0	M0	N0	T3	NX
155	tgct_tcga	TCGA-ZM-AA0N	TCGA-ZM-AA0N-01	44.0	Stage IB	M0	M0	N0	T2	NX

5 rows × 88 columns

In [6]:

```
df.shape
```

Out[6]:

```
(156, 88)
```

In [7]:

```
df.columns
```

Out[7]:

```
Index(['Study ID', 'Patient ID', 'Sample ID', 'Diagnosis Age',
       'Neoplasm Disease Stage American Joint Committee on Cancer Code',
       'Neoplasm American Joint Committee on Cancer Clinical Distant Metastasis M Stage',
       'American Joint Committee on Cancer Metastasis Stage Code',
       'American Joint Committee on Cancer Lymph Node Stage Code.1',
       'American Joint Committee on Cancer Lymph Node Stage Code',
       'Neoplasm Disease Lymph Node Stage American Joint Committee on Cancer Code',
       'Neoplasm Disease Stage American Joint Committee on Cancer Code.1',
       'American Joint Committee on Cancer Publication Version Type',
       'American Joint Committee on Cancer Tumor Stage Code', 'Cancer Type',
       'Cancer Type Detailed', 'Days to Sample Collection.',
       'Last Alive Less Initial Pathologic Diagnosis Date Calculated Day Value',
       'Days to post orchi serum test', 'Days to pre orchi serum test',
       'Disease Free (Months)', 'Disease Free Status', 'Disease code',
       'Ethnicity Category',
       'Lymphomatous Extranodal Site Involvement Indicator',
       'Family History Cancer Relationship', 'Family history other cancer',
       'Family history testicular cancer', 'First treatment success',
       'Form completion date', 'Fraction Genome Altered',
       'Neoplasm Histologic Type Name', 'Histologic diagnosis percent',
       'History fertility', 'History hypospadias',
       'Neoadjuvant Therapy Type Administered Prior To Resection Text',
       'History of undescended testis', 'Prior Cancer Diagnosis Occurence',
       'ICD-10 Classification',
       'International Classification of Diseases for Oncology, Third Edition ICD-O-3 Histol
ogy Code',
       'International Classification of Diseases for Oncology, Third Edition ICD-O-3 Site C
ode',
       'Igcccg stage', 'Informed consent verified',
       'Year Cancer Initial Diagnosis', 'Intratubular germ cell neoplasm',
       'Is FFPE', 'Primary Tumor Laterality',
       'Lymphovascular invasion present',
       'First Pathologic Diagnosis Biospecimen Acquisition Method Type',
       'Molecular test result', 'Mutation Count',
       'New Neoplasm Event Post Initial Therapy Indicator', 'Oct embedded',
       'Oncotree Code', 'Overall Survival (Months)', 'Overall Survival Status',
       'Other Patient ID', 'Other Sample ID', 'Pathology Report File Name',
       'Pathology report uuid',
       'Adjuvant Postoperative Pharmaceutical Therapy Administered Indicator',
       'Postoperative tx', 'Post orchi afp', 'Post orchi hcg',
       'Post orchi ldh', 'Post orchi lymph node dissection', 'Pre orchi afp',
       'Pre orchi hcg', 'Pre orchi ldh',
       'Tissue Prospective Collection Indicator', 'Race Category',
       'Did patient start adjuvant postoperative radiotherapy?',
       'Relation testicular cancer', 'Relative family cancer hx text',
       'Tissue Retrospective Collection Indicator',
       'Number of Samples Per Patient', 'Sample Initial Weight', 'Sample Type',
       'Sample type id', 'Serum markers', 'Sex', 'Tumor Tissue Site',
       'Somatic Status', 'Testis tumor macroextent',
       'Testis tumor microextent', 'Tissue Source Site', 'TMB (nonsynonymous)',
       'Person Neoplasm Status', 'Vial number'],
      dtype='object')
```

In [8]:

```
df.duplicated().sum()
```

Out[8]:

```
0
```

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
Study ID                                                      0
Patient ID                                                    0
Sample ID                                                     0
Diagnosis Age                                                17
Neoplasm Disease Stage American Joint Committee on Cancer Code    26
                                                            ...
Testis tumor microextent                                    118
Tissue Source Site                                           17
TMB (nonsynonymous)                                           1
Person Neoplasm Status                                       21
Vial number                                                   6
Length: 88, dtype: int64
```

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 156 entries, 0 to 155
Data columns (total 88 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Study ID
156 non-null    object
 1   Patient ID
156 non-null    object
 2   Sample ID
156 non-null    object
 3   Diagnosis Age
139 non-null    float64
 4   Neoplasm Disease Stage American Joint Committee on Cancer Code
130 non-null    object
 5   Neoplasm American Joint Committee on Cancer Clinical Distant Metastasis M Stage
133 non-null    object
 6   American Joint Committee on Cancer Metastasis Stage Code
124 non-null    object
 7   American Joint Committee on Cancer Lymph Node Stage Code.1
131 non-null    object
 8   American Joint Committee on Cancer Lymph Node Stage Code
118 non-null    object
 9   Neoplasm Disease Lymph Node Stage American Joint Committee on Cancer Code
129 non-null    object
 10  Neoplasm Disease Stage American Joint Committee on Cancer Code.1
132 non-null    object
 11  American Joint Committee on Cancer Publication Version Type
139 non-null    object
 12  American Joint Committee on Cancer Tumor Stage Code
139 non-null    object
 13  Cancer Type
156 non-null    object
 14  Cancer Type Detailed
156 non-null    object
 15  Days to Sample Collection.
134 non-null    float64
 16  Last Alive Less Initial Pathologic Diagnosis Date Calculated Day Value
139 non-null    float64
 17  Days to post orchi serum test
123 non-null    float64
 18  Days to pre orchi serum test
130 non-null    float64
 19  Disease Free (Months)
137 non-null    float64
 20  Disease Free Status
137 non-null    object
 21  Disease code
0 non-null      float64
 22  Ethnicity Category
128 non-null    object
 23  Lymphomatous Extranodal Site Involvement Indicator
0 non-null      float64
 24  Family History Cancer Relationship
49 non-null     object
 25  Family history other cancer
110 non-null    object
 26  Family history testicular cancer
122 non-null    object
 27  First treatment success
119 non-null    object
 28  Form completion date
139 non-null    object
 29  Fraction Genome Altered
156 non-null    float64
 30  Neoplasm Histologic Type Name
139 non-null    object
 31  Histologic diagnosis percent
```

```
139 non-null    object
 32  History fertility
112 non-null    object
 33  History hypospadias
131 non-null    object
 34  Neoadjuvant Therapy Type Administered Prior To Resection Text
139 non-null    object
 35  History of undescended testis
132 non-null    object
 36  Prior Cancer Diagnosis Occurence
139 non-null    object
 37  ICD-10 Classification
139 non-null    object
 38  International Classification of Diseases for Oncology, Third Edition ICD-O-3 Histology
Code  139 non-null    object
 39  International Classification of Diseases for Oncology, Third Edition ICD-O-3 Site Code
139 non-null    object
 40  Igcccg stage
43 non-null     object
 41  Informed consent verified
139 non-null    object
 42  Year Cancer Initial Diagnosis
139 non-null    float64
 43  Intratubular germ cell neoplasm
130 non-null    object
 44  Is FFPE
150 non-null    object
 45  Primary Tumor Laterality
139 non-null    object
 46  Lymphovascular invasion present
135 non-null    object
 47  First Pathologic Diagnosis Biospecimen Acquisition Method Type
139 non-null    object
 48  Molecular test result
120 non-null    object
 49  Mutation Count
155 non-null    float64
 50  New Neoplasm Event Post Initial Therapy Indicator
134 non-null    object
 51  Oct embedded
150 non-null    object
 52  Oncotree Code
156 non-null    object
 53  Overall Survival (Months)
139 non-null    float64
 54  Overall Survival Status
139 non-null    object
 55  Other Patient ID
139 non-null    object
 56  Other Sample ID
150 non-null    object
 57  Pathology Report File Name
150 non-null    object
 58  Pathology report uuid
150 non-null    object
 59  Adjuvant Postoperative Pharmaceutical Therapy Administered Indicator
128 non-null    object
 60  Postoperative tx
132 non-null    object
 61  Post orchi afp
118 non-null    float64
 62  Post orchi hcg
68 non-null     float64
 63  Post orchi ldh
115 non-null    float64
 64  Post orchi lymph node dissection
128 non-null    object
 65  Pre orchi afp
127 non-null    float64
 66  Pre orchi hcg
101 non-null    float64
```

```
 67  Pre orchi ldh
119 non-null    float64
 68  Tissue Prospective Collection Indicator
137 non-null    object
 69  Race Category
134 non-null    object
 70  Did patient start adjuvant postoperative radiotherapy?
127 non-null    object
 71  Relation testicular cancer
15 non-null     object
 72  Relative family cancer hx text
49 non-null     object
 73  Tissue Retrospective Collection Indicator
139 non-null    object
 74  Number of Samples Per Patient
156 non-null    int64
 75  Sample Initial Weight
148 non-null    float64
 76  Sample Type
156 non-null    object
 77  Sample type id
156 non-null    int64
 78  Serum markers
137 non-null    object
 79  Sex
139 non-null    object
 80  Tumor Tissue Site
139 non-null    object
 81  Somatic Status
156 non-null    object
 82  Testis tumor macroextent
124 non-null    object
 83  Testis tumor microextent
38 non-null     object
 84  Tissue Source Site
139 non-null    object
 85  TMB (nonsynonymous)
155 non-null    float64
 86  Person Neoplasm Status
135 non-null    object
 87  Vial number
150 non-null    object
dtypes: float64(20), int64(2), object(66)
memory usage: 107.4+ KB
```

In [11]:

```
df.describe()
```

Out[11]:

	Diagnosis Age	Days to Sample Collection.	Last Alive Less Initial Pathologic Diagnosis Date Calculated Day Value	Days to post orchi serum test	Days to pre orchi serum test	Disease Free (Months)	Disease code	Lymphomatous Extranodal Site Involvement Indicator	Fra Ge Al
count	139.000000	134.000000	139.0	123.000000	130.000000	137.000000	0.0	0.0	156.0(
mean	31.870504	2064.238806	0.0	84.837398	-5.853846	54.468467	NaN	NaN	0.3(
std	9.188710	1981.344926	0.0	189.093280	42.567432	59.737355	NaN	NaN	0.2(
min	14.000000	34.000000	0.0	-61.000000	-310.000000	0.100000	NaN	NaN	0.0(
25%	26.000000	507.000000	0.0	11.500000	-7.000000	14.320000	NaN	NaN	0.1(
50%	31.000000	1480.000000	0.0	24.000000	-4.000000	27.860000	NaN	NaN	0.3(
75%	37.000000	2654.500000	0.0	56.500000	-3.000000	73.290000	NaN	NaN	0.5(
max	67.000000	7751.000000	0.0	1455.000000	361.000000	232.620000	NaN	NaN	0.8(

8 rows × 22 columns

In [12]:

```
new_column_names = [
    'study_id', 'patient_id', 'sample_id', 'diagnosis_age', 'neoplasm_stage_code',
    'neoplasm_metastasis_m_stage', 'metastasis_stage_code', 'lymph_node_stage_code_1',
    'lymph_node_stage_code', 'lymph_node_neoplasm_stage', 'neoplasm_stage_code_1',
    'publication_version_type', 'tumor_stage_code', 'cancer_type', 'cancer_type_detailed',
    'days_to_sample_collection', 'days_to_diagnosis_calculation', 'days_to_post_orchi_serum_test',
    'days_to_pre_orchi_serum_test', 'disease_free_months', 'disease_free_status', 'disease_code',
    'ethnicity_category', 'lymphomatous_extranodal_involvement', 'family_cancer_relationship',
    'family_cancer_other', 'family_cancer_testicular', 'first_treatment_success',
    'form_completion_date', 'fraction_genome_altered', 'histologic_type_name',
    'histologic_diagnosis_percent', 'history_fertility', 'history_hypospadias',
    'neoadjuvant_therapy_text', 'history_undescended_testis', 'prior_cancer_diagnosis_occurrence',
    'icd_10_classification', 'icd_o_3_histology_code', 'icd_o_3_site_code', 'igcccg_stage',
    'informed_consent_verified', 'year_diagnosis', 'intratubular_germ_cell_neoplasm',
    'is_ffpe', 'primary_tumor_laterality', 'lymphovascular_invasion_present',
    'first_pathologic_acquisition_method', 'molecular_test_result', 'mutation_count',
    'new_neoplasm_post_therapy', 'oct_embedded', 'oncotree_code', 'overall_survival_months',
    'overall_survival_status', 'other_patient_id', 'other_sample_id', 'pathology_report_file_name',
    'pathology_report_uuid', 'adjuvant_pharmaceutical_therapy', 'postoperative_tx',
    'post_orchi_afp', 'post_orchi_hcg', 'post_orchi_ldh', 'post_orchi_lymph_node_dissection',
    'pre_orchi_afp', 'pre_orchi_hcg', 'pre_orchi_ldh', 'tissue_prospective_collection',
    'race_category', 'start_adjuvant_radiotherapy', 'relation_testicular_cancer',
    'relative_family_cancer_text', 'tissue_retrospective_collection', 'samples_per_patient',
    'sample_initial_weight', 'sample_type', 'sample_type_id', 'serum_markers', 'sex',
    'tumor_tissue_site', 'somatic_status', 'testis_tumor_macroextent', 'testis_tumor_microextent',
    'tissue_source_site', 'tmb_nonsynonymous', 'person_neoplasm_status', 'vial_number'
]
```

In [13]:

```
df.columns = new_column_names
```

In [14]:

```
df = df.drop(['disease_code', 'lymphomatous_extranodal_involvement'], axis = 1)
```

In [15]:

```
object_columns = df.select_dtypes(include='object').columns.tolist()
numerical_columns = df.select_dtypes(include=['int', 'float']).columns.tolist()

print("Object columns:", object_columns)
print('\n')
print("Numerical columns:", numerical_columns)
```

```
Object columns: ['study_id', 'patient_id', 'sample_id', 'neoplasm_stage_code', 'neoplasm_me
tastasis_m_stage', 'metastasis_stage_code', 'lymph_node_stage_code_1', 'lymph_node_stage_co
de', 'lymph_node_neoplasm_stage', 'neoplasm_stage_code_1', 'publication_version_type', 'tum
or_stage_code', 'cancer_type', 'cancer_type_detailed', 'disease_free_status', 'ethnicity_ca
tegory', 'family_cancer_relationship', 'family_cancer_other', 'family_cancer_testicular',
'first_treatment_success', 'form_completion_date', 'histologic_type_name', 'histologic_diag
nosis_percent', 'history_fertility', 'history_hypospadias', 'neoadjuvant_therapy_text', 'hi
story_undescended_testis', 'prior_cancer_diagnosis_occurrence', 'icd_10_classification', 'i
cd_o_3_histology_code', 'icd_o_3_site_code', 'igcccg_stage', 'informed_consent_verified',
'intratubular_germ_cell_neoplasm', 'is_ffpe', 'primary_tumor_laterality', 'lymphovascular_i
nvasion_present', 'first_pathologic_acquisition_method', 'molecular_test_result', 'new_neop
lasm_post_therapy', 'oct_embedded', 'oncotree_code', 'overall_survival_status', 'other_pati
ent_id', 'other_sample_id', 'pathology_report_file_name', 'pathology_report_uuid', 'adjuvan
t_pharmaceutical_therapy', 'postoperative_tx', 'post_orchi_lymph_node_dissection', 'tissue_
prospective_collection', 'race_category', 'start_adjuvant_radiotherapy', 'relation_testicul
ar_cancer', 'relative_family_cancer_text', 'tissue_retrospective_collection', 'sample_typ
e', 'serum_markers', 'sex', 'tumor_tissue_site', 'somatic_status', 'testis_tumor_macroexten
t', 'testis_tumor_microextent', 'tissue_source_site', 'person_neoplasm_status', 'vial_numbe
r']


Numerical columns: ['diagnosis_age', 'days_to_sample_collection', 'days_to_diagnosis_calcul
ation', 'days_to_post_orchi_serum_test', 'days_to_pre_orchi_serum_test', 'disease_free_mont
hs', 'fraction_genome_altered', 'year_diagnosis', 'mutation_count', 'overall_survival_month
s', 'post_orchi_afp', 'post_orchi_hcg', 'post_orchi_ldh', 'pre_orchi_afp', 'pre_orchi_hcg',
'pre_orchi_ldh', 'samples_per_patient', 'sample_initial_weight', 'sample_type_id', 'tmb_non
synonymous']
```

In [16]:

```
def identify_numeric_type(df, column_name):
    unique_values_count = len(df[column_name].unique())
    if unique_values_count < 10:
        return 'Discrete'
    else:
        return 'Continuous'
```

In [17]:

```
discrete_numeric_columns = []
continuous_numeric_columns = []
```

In [18]:

```
for column in df.columns:
    if df[column].dtype == 'float64' or df[column].dtype == 'int64':
        column_type = identify_numeric_type(df, column)
        if column_type == 'Discrete':
            discrete_numeric_columns.append(column)
        elif column_type == 'Continuous':
            continuous_numeric_columns.append(column)
```

In [19]:

```
print('Discrete Numeric Columns:', discrete_numeric_columns)
print('\n')
print('Continuous Numeric Columns:', continuous_numeric_columns)
```

```
Discrete Numeric Columns: ['days_to_diagnosis_calculation', 'samples_per_patient', 'sample_
type_id']


Continuous Numeric Columns: ['diagnosis_age', 'days_to_sample_collection', 'days_to_post_or
chi_serum_test', 'days_to_pre_orchi_serum_test', 'disease_free_months', 'fraction_genome_al
tered', 'year_diagnosis', 'mutation_count', 'overall_survival_months', 'post_orchi_afp', 'p
ost_orchi_hcg', 'post_orchi_ldh', 'pre_orchi_afp', 'pre_orchi_hcg', 'pre_orchi_ldh', 'sampl
e_initial_weight', 'tmb_nonsynonymous']
```

In [20]:

```
def identify_object_categorical_type(df, column_name):
    unique_values_count = len(df[column_name].unique())
    if unique_values_count < 10:
        return 'Categorical'
    else:
        return 'Non Categorical'
```

In [21]:

```
categorical_columns = []
non_categorical_columns = []
```

In [22]:

```
for column in df.columns:
    if df[column].dtype == 'object':
        column_type = identify_object_categorical_type(df, column)
        if column_type == 'Categorical':
            categorical_columns.append(column)
        elif column_type == 'Non Categorical':
            non_categorical_columns.append(column)
```

In [23]:

```
print('Categorical Columns:', categorical_columns)
print('\n')
print('Non-Categorical Columns:', non_categorical_columns)
```

```
Categorical Columns: ['study_id', 'neoplasm_metastasis_m_stage', 'metastasis_stage_code',
'lymph_node_stage_code_1', 'lymph_node_stage_code', 'lymph_node_neoplasm_stage', 'publicati
on_version_type', 'tumor_stage_code', 'cancer_type', 'cancer_type_detailed', 'disease_free_
status', 'ethnicity_category', 'family_cancer_other', 'family_cancer_testicular', 'first_tr
eatment_success', 'history_fertility', 'history_hypospadias', 'neoadjuvant_therapy_text',
'history_undescended_testis', 'prior_cancer_diagnosis_occurrence', 'icd_10_classification',
'icd_o_3_histology_code', 'icd_o_3_site_code', 'igcccg_stage', 'informed_consent_verified',
'intratubular_germ_cell_neoplasm', 'is_ffpe', 'primary_tumor_laterality', 'lymphovascular_i
nvasion_present', 'first_pathologic_acquisition_method', 'new_neoplasm_post_therapy', 'oct_
embedded', 'oncotree_code', 'overall_survival_status', 'adjuvant_pharmaceutical_therapy',
'postoperative_tx', 'post_orchi_lymph_node_dissection', 'tissue_prospective_collection', 'r
ace_category', 'start_adjuvant_radiotherapy', 'relation_testicular_cancer', 'tissue_retrosp
ective_collection', 'sample_type', 'serum_markers', 'sex', 'tumor_tissue_site', 'somatic_st
atus', 'testis_tumor_macroextent', 'testis_tumor_microextent', 'person_neoplasm_status', 'v
ial_number']


Non-Categorical Columns: ['patient_id', 'sample_id', 'neoplasm_stage_code', 'neoplasm_stage
_code_1', 'family_cancer_relationship', 'form_completion_date', 'histologic_type_name', 'hi
stologic_diagnosis_percent', 'molecular_test_result', 'other_patient_id', 'other_sample_i
d', 'pathology_report_file_name', 'pathology_report_uuid', 'relative_family_cancer_text',
'tissue_source_site']
```

In [24]:

```
for col in continuous_numeric_columns:
    df[col].fillna(df[col].mean(), inplace=True)
```

In [25]:

```
for col in discrete_numeric_columns:
    if df[col].isnull().sum() > 0:
        mode_val = df[col].mode()
        if not mode_val.empty:
            df[col].fillna(mode_val.iloc[0], inplace=True)
```

In [26]:

```
for col in object_columns:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

In [27]:

```
columns_with_nulls = df[continuous_numeric_columns].columns[df[continuous_numeric_columns].isnull().any()
print("Column Names with Null Values:")
print(columns_with_nulls)
```

```
Column Names with Null Values:
Index([], dtype='object')
```

In [28]:

```
columns_with_nulls = df[discrete_numeric_columns].columns[df[discrete_numeric_columns].isnull().any()]
print("Column Names with Null Values:")
print(columns_with_nulls)
```

```
Column Names with Null Values:
Index([], dtype='object')
```

In [29]:

```
columns_with_nulls = df[categorical_columns].columns[df[categorical_columns].isnull().any()]
print("Column Names with Null Values:")
print(columns_with_nulls)
```

```
Column Names with Null Values:
Index([], dtype='object')
```

In [30]:

```
columns_with_nulls = df[non_categorical_columns].columns[df[non_categorical_columns].isnull().any()]
print("Column Names with Null Values:")
print(columns_with_nulls)
```

```
Column Names with Null Values:
Index([], dtype='object')
```

In [31]:

```
df.isnull().sum()
```

Out[31]:

```
study_id                    0
patient_id                  0
sample_id                   0
diagnosis_age               0
neoplasm_stage_code         0
                           ..
testis_tumor_microextent    0
tissue_source_site          0
tmb_nonsynonymous           0
person_neoplasm_status      0
vial_number                 0
Length: 86, dtype: int64
```

In [32]:

```
df.nunique()
```

Out[32]:

```
study_id                    1
patient_id                150
sample_id                 156
diagnosis_age              39
neoplasm_stage_code        12
                         ...
testis_tumor_microextent    4
tissue_source_site         15
tmb_nonsynonymous          52
person_neoplasm_status      2
vial_number                 2
Length: 86, dtype: int64
```

In [33]:

```
for i in categorical_columns:
    print(i)
    print(df[i].unique())
    print('\n')
```

```
study_id
['tgct_tcga']


neoplasm_metastasis_m_stage
['M0' 'M1b' 'M1a' 'M1']


metastasis_stage_code
['M0' 'M1' 'M1b' 'M1a']


lymph_node_stage_code_1
['N0' 'N1' 'N2' 'NX' 'N3']


lymph_node_stage_code
['T1' 'T2' 'T3']
```

In [34]:

```
for i in categorical_columns:
    print(i)
    print(df[i].value_counts())
    print('\n')
```

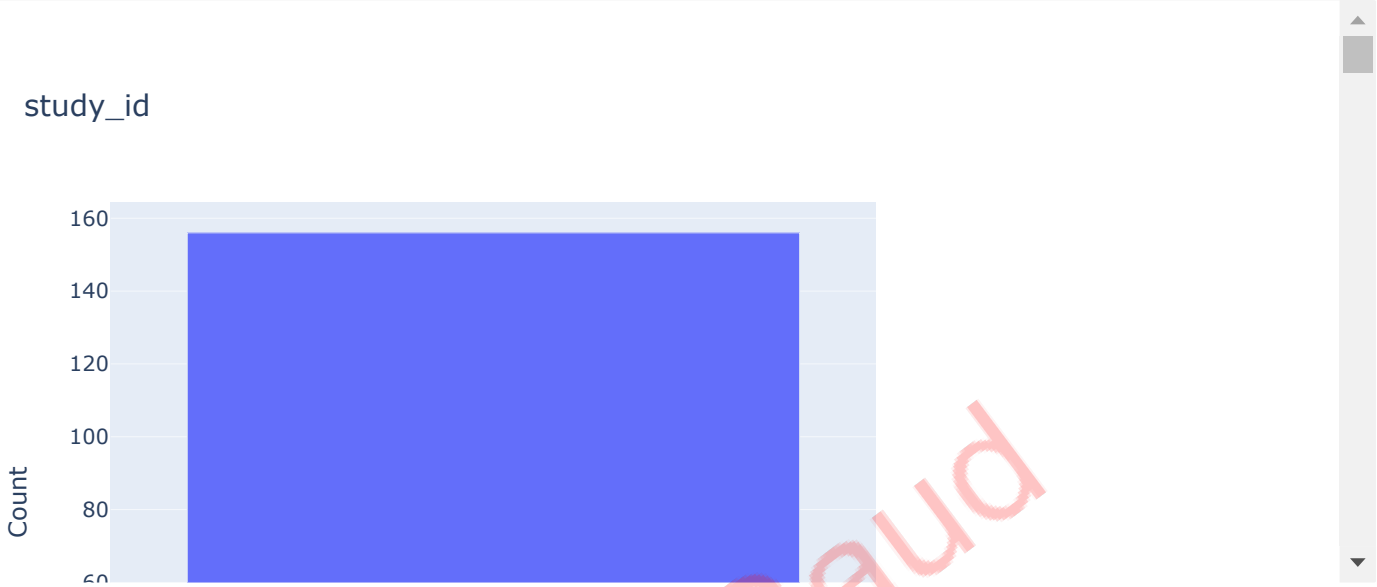
```
study_id
tgct_tcga    156
Name: study_id, dtype: int64


neoplasm_metastasis_m_stage
M0     148
M1a      4
M1b      3
M1       1
Name: neoplasm_metastasis_m_stage, dtype: int64


metastasis_stage_code
M0     152
M1       2
M1b      1
M1a      1
Name: metastasis_stage_code, dtype: int64
```

In [35]:

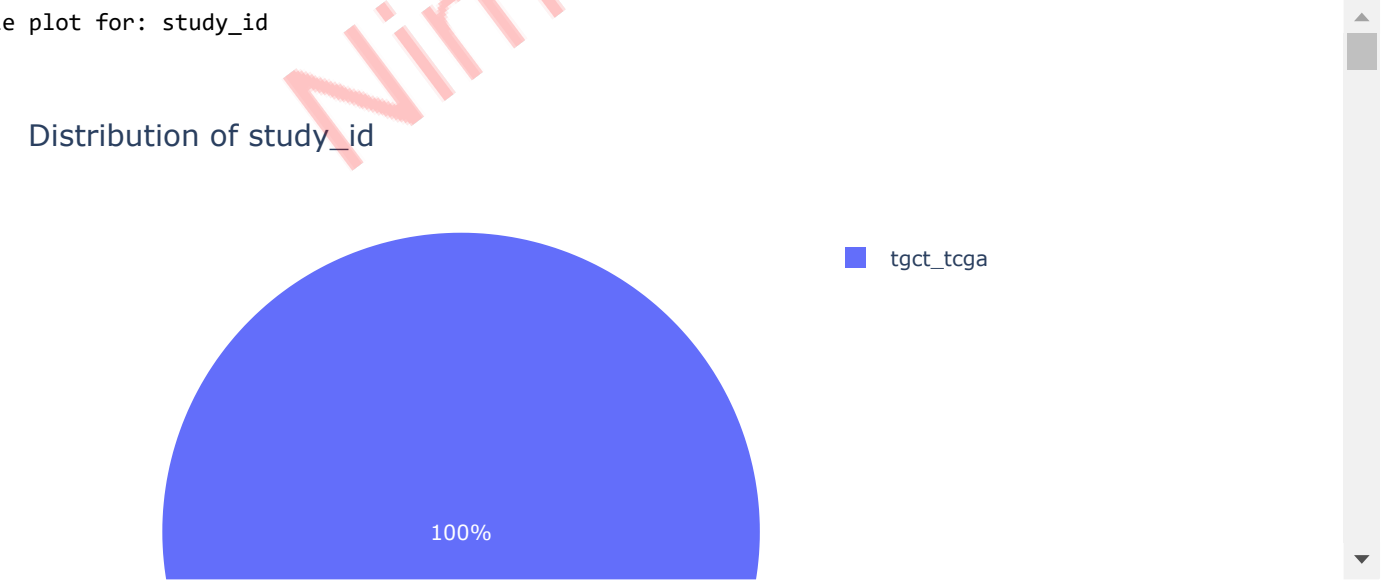
```
for i in categorical_columns:
    fig = go.Figure(data=[go.Bar(x=df[i].value_counts().index,
                                 y=df[i].value_counts())])
    fig.update_layout(
        title=i,
        xaxis_title=i,
        yaxis_title="Count")
    fig.show()
```



In [36]:

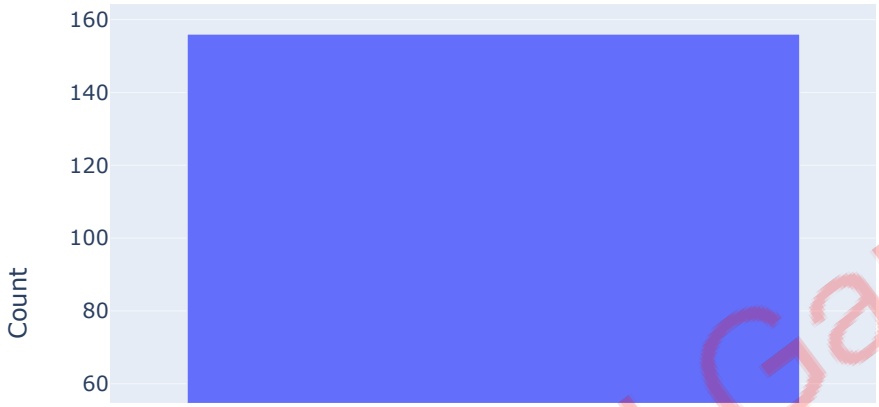
```
for i in categorical_columns:
    print('Pie plot for:', i)
    fig = px.pie(df, names=i, title='Distribution of ' + i)
    fig.show()
    print('\n')
```

```
Pie plot for: study_id
```



In [37]:

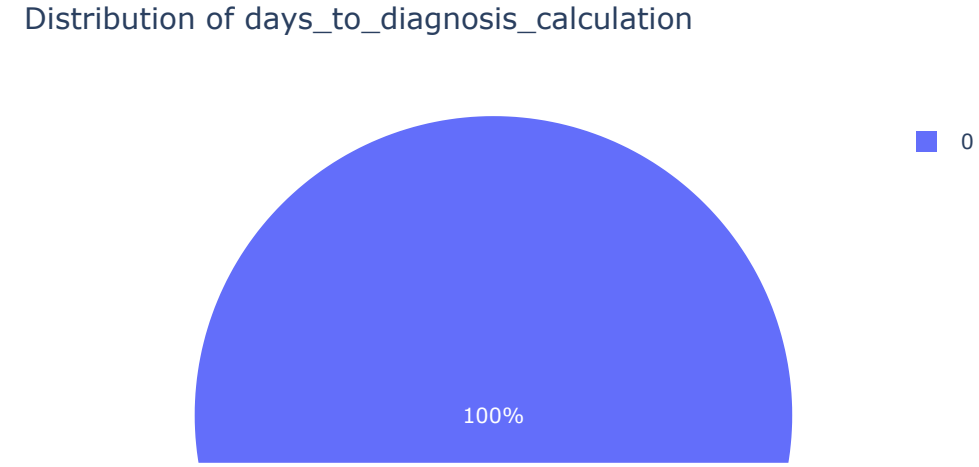
```
for col in discrete_numeric_columns:
    if col in df.columns:
        fig = go.Figure(data=[go.Bar(x=df[col].value_counts().index,
                                     y=df[col].value_counts())])
        fig.update_layout(
            title=col,
            xaxis_title=col,
            yaxis_title="Count")
        fig.show()
    else:
        print(f"Column '{col}' not found in the DataFrame.")
```



In [38]:

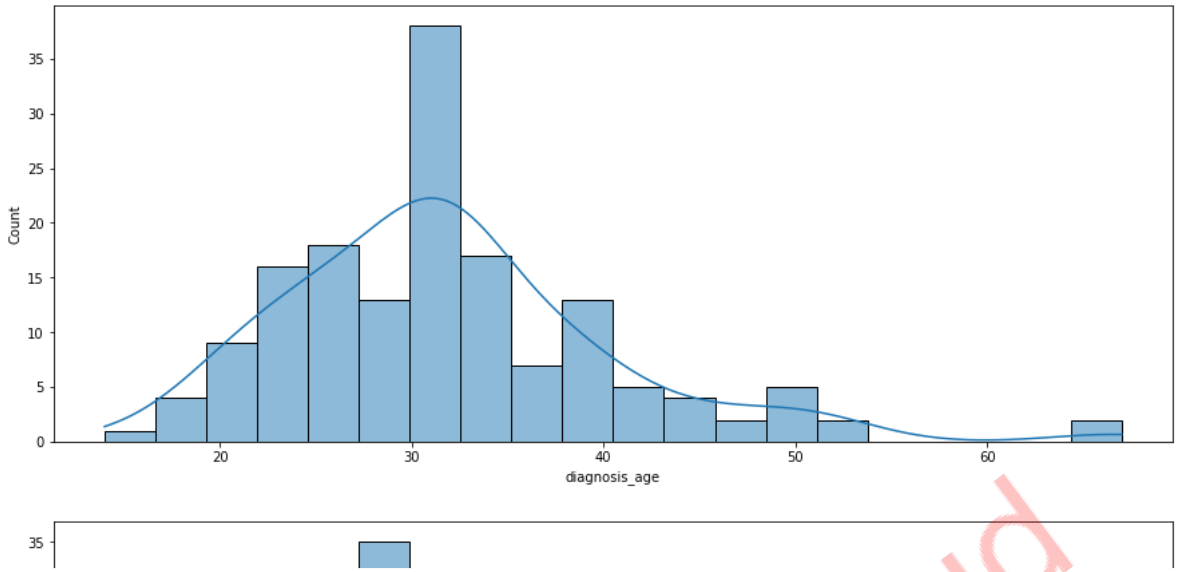
```
for i in discrete_numeric_columns:
    print('Pie plot for:', i)
    fig = px.pie(df, names=i, title='Distribution of ' + i)
    fig.show()
    print('\n')
```

```
Pie plot for: days_to_diagnosis_calculation
```



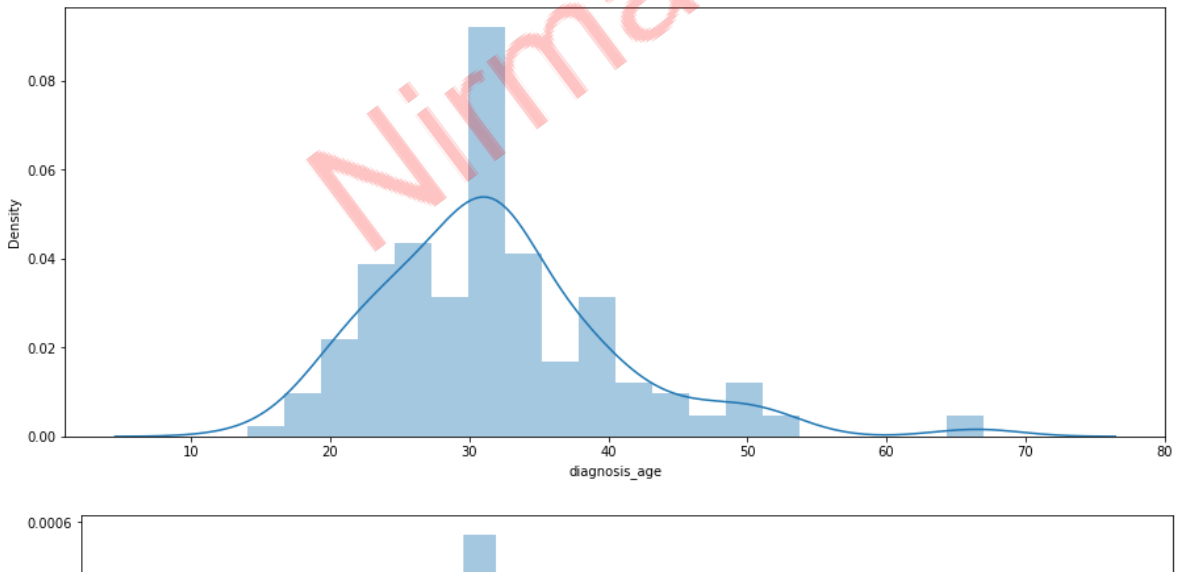
In [39]:

```
for i in continuous_numeric_columns:
    plt.figure(figsize=(15,6))
    sns.histplot(df[i], kde = True, bins = 20, palette = 'hls')
    plt.xticks(rotation = 0)
    plt.show()
```



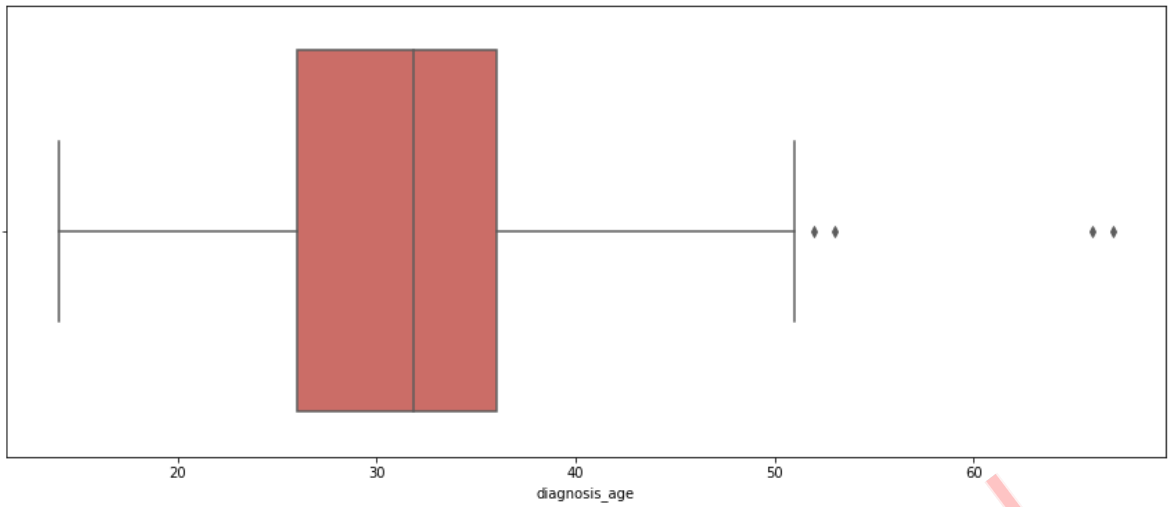
In [40]:

```
for i in continuous_numeric_columns:
    plt.figure(figsize=(15,6))
    sns.distplot(df[i], kde = True, bins = 20)
    plt.xticks(rotation = 0)
    plt.show()
```



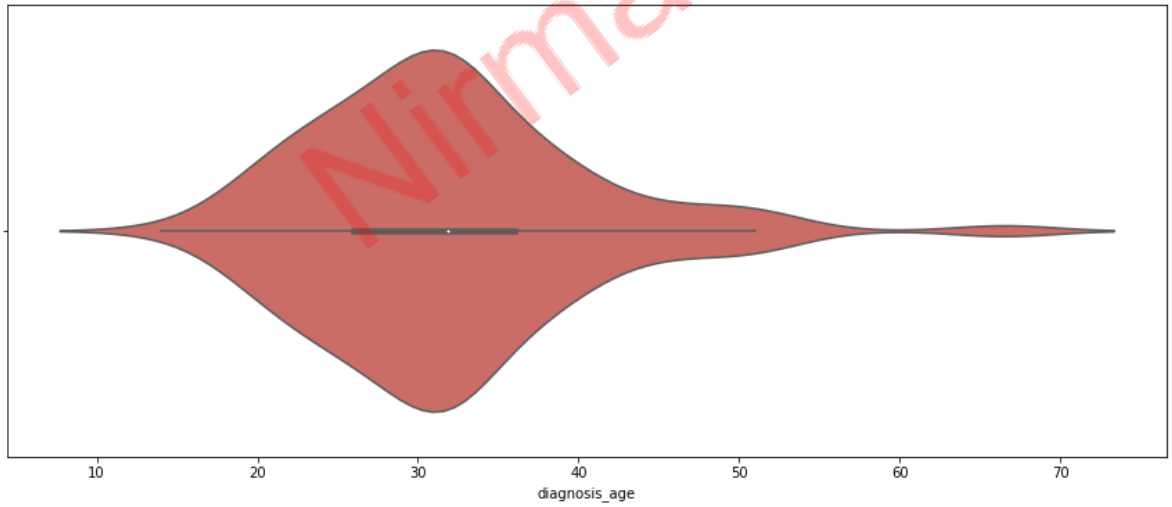
In [41]:

```
for i in continuous_numeric_columns:
    plt.figure(figsize=(15,6))
    sns.boxplot(df[i], data=df, palette='hls')
    plt.xticks(rotation = 0)
    plt.show()
```



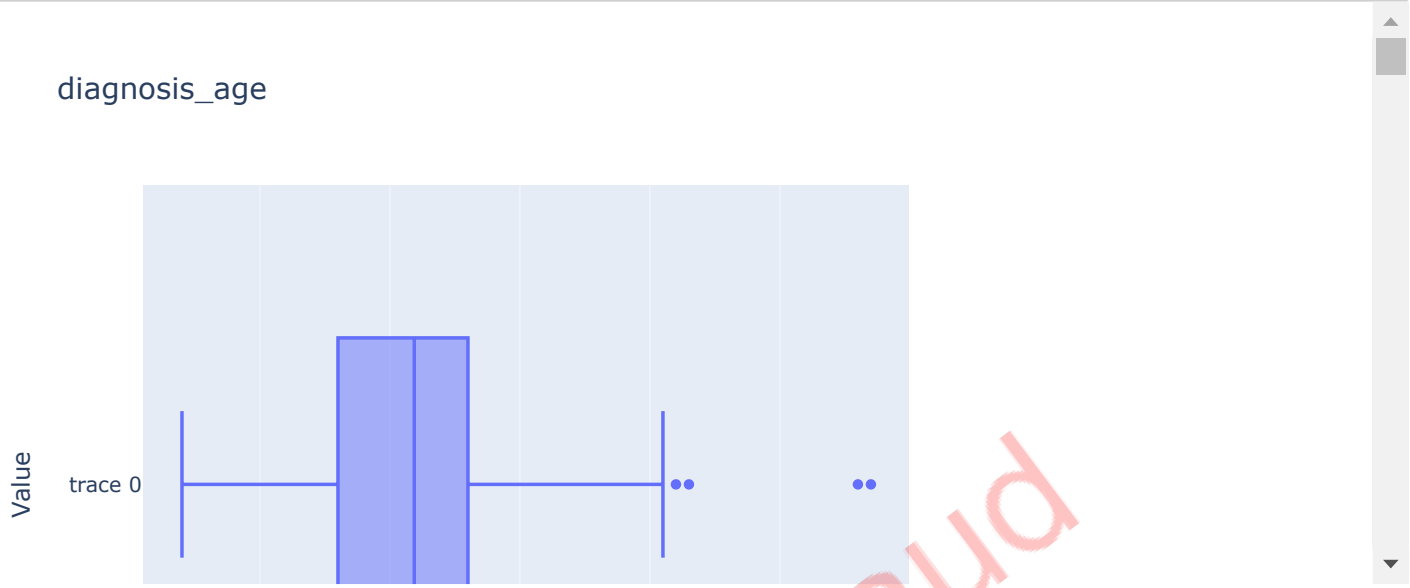
In [42]:

```
for i in continuous_numeric_columns:
    plt.figure(figsize=(15,6))
    sns.violinplot(df[i], data=df, palette='hls')
    plt.xticks(rotation = 0)
    plt.show()
```



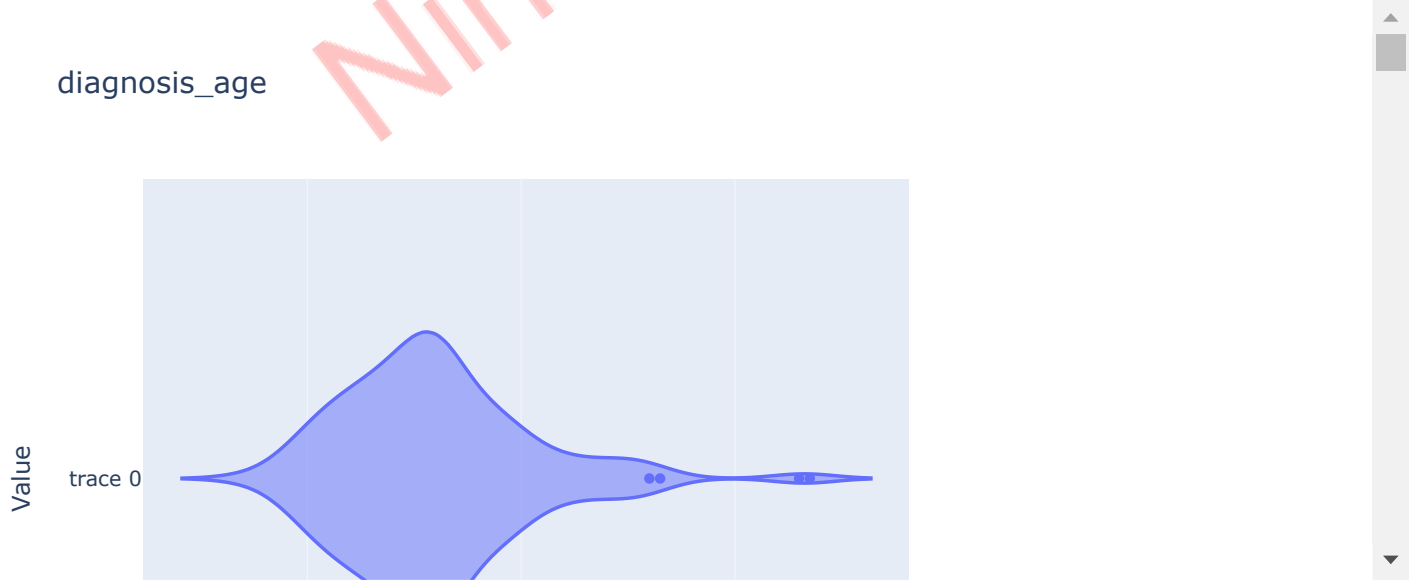
In [43]:

```
for i in continuous_numeric_columns:
    fig = go.Figure(data=[go.Box(x=df[i])])
    fig.update_layout(
        title=i,
        xaxis_title=i,
        yaxis_title="Value")
    fig.show()
```



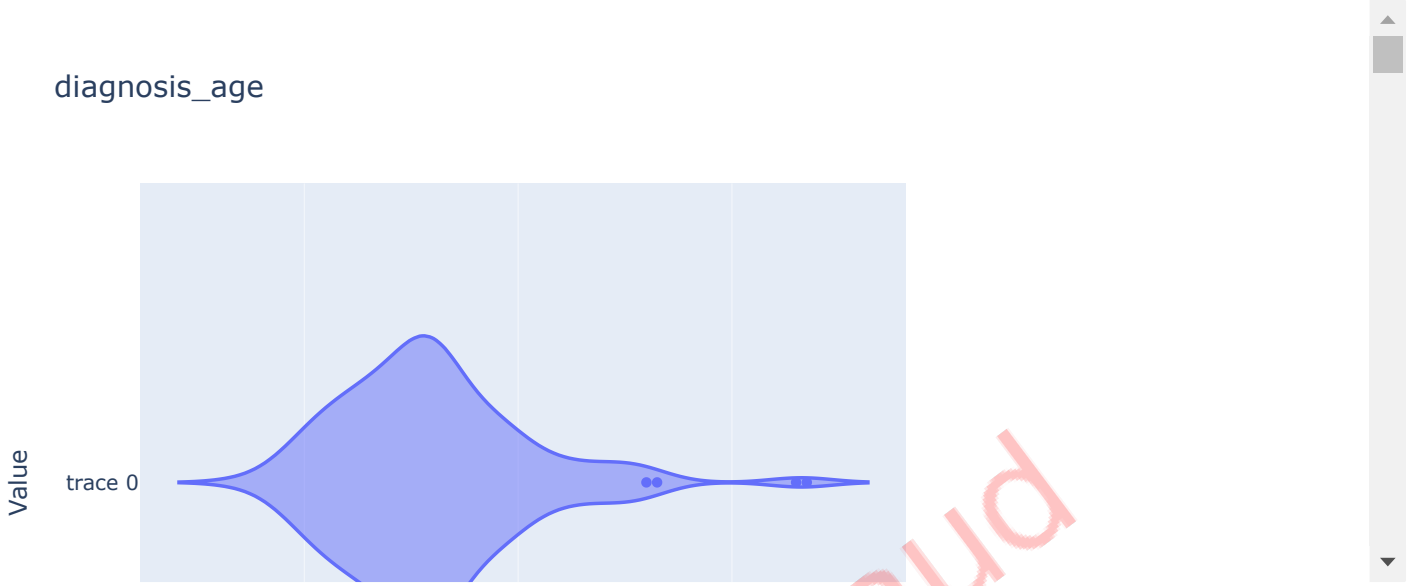
In [44]:

```
for i in continuous_numeric_columns:
    fig = go.Figure(data=[go.Violin(x=df[i])])
    fig.update_layout(
        title=i,
        xaxis_title=i,
        yaxis_title="Value")
    fig.show()
```



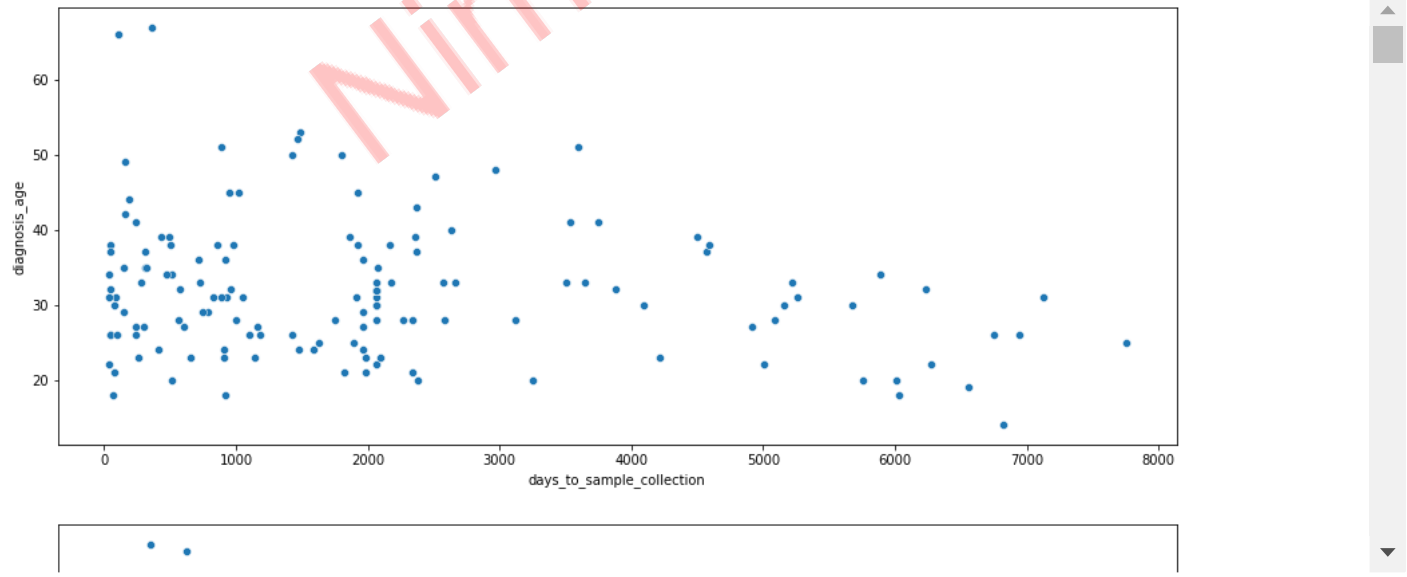
In [45]:

```
for i in continuous_numeric_columns:
    fig = go.Figure(data=[go.Violin(x=df[i])])
    fig.update_layout(
        title=i,
        xaxis_title=i,
        yaxis_title="Value")
    fig.show()
```



In [46]:

```
for i in continuous_numeric_columns:
    for j in continuous_numeric_columns:
        if i != j:
            plt.figure(figsize=(15,6))
            sns.scatterplot(x = df[j], y = df[i], data = df, palette = 'hls')
            plt.show()
```



In [47]:

```
df1 = df.copy()
```

In [48]:

```
df1 = pd.get_dummies(df1, columns=categorical_columns, drop_first=True)
```

In [49]:

```
df1 = df1.drop(non_categorical_columns, axis = 1)
```

In [50]:

```
df1 = df1.drop('days_to_diagnosis_calculation', axis = 1)
```

In [51]:

```
corr = df1.corr()
```

In [52]:

```
corr
```

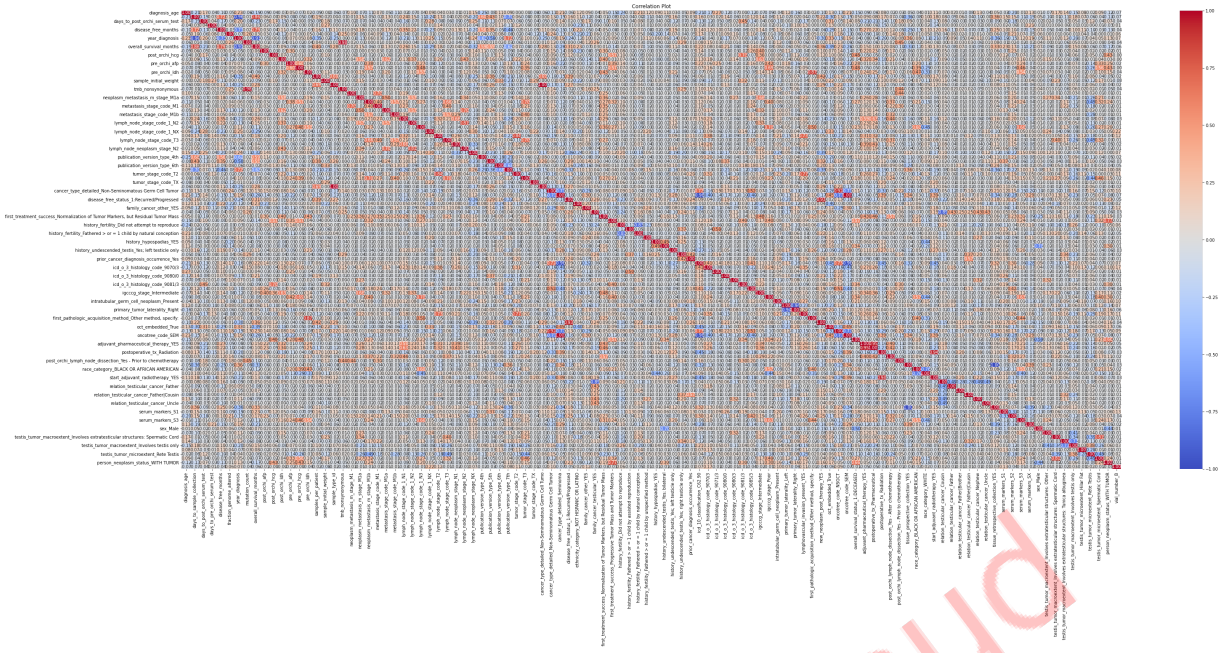
Out[52]:

	diagnosis_age	days_to_sample_collection	days_to_post_orchi_serum_test	day
diagnosis_age	1.000000	-0.209098	0.170381	
days_to_sample_collection	-0.209098	1.000000	-0.186470	
days_to_post_orchi_serum_test	0.170381	-0.186470	1.000000	
days_to_pre_orchi_serum_test	-0.041121	0.026037	0.041579	
disease_free_months	-0.099871	0.716565	-0.169487	
...	...	...	...	
testis_tumor_microextent_Hilar Fat	0.084849	0.073793	-0.034025	
testis_tumor_microextent_Rete Testis	-0.015460	-0.063776	0.115261	
testis_tumor_microextent_Spermatic Cord	0.088512	-0.038416	-0.038149	
person_neoplasm_status_WITH TUMOR	-0.123606	-0.129608	0.074320	
vial_number_B	-0.073148	-0.072760	-0.036427	

108 rows × 108 columns

In [53]:

```
plt.figure(figsize=(50, 20))
sns.heatmap(corr, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Plot')
plt.show()
```



In [54]:

```
relevant_features = corr[corr['overall_survival_months'] > 0.3]
```

In [55]:

```
relevant_features_list = relevant_features.index.tolist()
print("Features with correlation greater than 0.3 with 'Overall Survival (Months)':")
print(relevant_features_list)
```

```
Features with correlation greater than 0.3 with 'Overall Survival (Months)':
['days_to_sample_collection', 'disease_free_months', 'overall_survival_months', 'samples_pe
r_patient', 'publication_version_type_4th', 'publication_version_type_5th', 'disease_free_s
tatus_1:Recurred/Progressed', 'new_neoplasm_post_therapy_YES', 'tissue_retrospective_collec
tion_YES']
```

In [56]:

```
relevant_features = ['days_to_sample_collection', 'disease_free_months', 'overall_survival_months',
                     'samples_per_patient', 'publication_version_type_4th', 'publication_version_type_5th
                     'disease_free_status_1:Recurred/Progressed', 'new_neoplasm_post_therapy_YES',
                     'tissue_retrospective_collection_YES']
```

In [57]:

```
relevant_df = df1[relevant_features]
```

In [58]:

```
relevant_df
```

Out[58]:

	days_to_sample_collection	disease_free_months	overall_survival_months	samples_per_patient	publication_ve
0	829.0	4.70	20.30	1	
1	858.0	41.36	41.36	1	
2	996.0	46.09	46.09	1	
3	2069.0	76.05	76.05	1	
4	3119.0	26.61	114.68	1	
...	...	...	...	...	
151	472.0	27.86	27.86	1	
152	435.0	26.64	26.64	1	
153	316.0	22.37	22.37	1	
154	1422.0	57.03	57.03	1	
155	191.0	16.10	20.83	1	

156 rows × 9 columns

In [59]:

```
features_to_transform = ['days_to_sample_collection', 'disease_free_months', 'overall_survival_months']
```

In [60]:

```
relevant_df[features_to_transform] = relevant_df[features_to_transform].apply(lambda x: np.log(x + 1))
```

In [61]:

```
print("DataFrame with log-transformed features:")
relevant_df
```

```
DataFrame with log-transformed features:
```

Out[61]:

	days_to_sample_collection	disease_free_months	overall_survival_months	samples_per_patient	publication_ve
0	6.721426	1.740466	3.058707	1	
1	6.755769	3.746205	3.746205	1	
2	6.904751	3.852061	3.852061	1	
3	7.635304	4.344455	4.344455	1	
4	8.045588	3.318178	4.750828	1	
...	...	...	...	...	
151	6.159095	3.362457	3.362457	1	
152	6.077642	3.319264	3.319264	1	
153	5.758902	3.151453	3.151453	1	
154	7.260523	4.060960	4.060960	1	
155	5.257495	2.839078	3.083285	1	

156 rows × 9 columns

In [62]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [63]:

```
predictors = ['days_to_sample_collection', 'disease_free_months', 'samples_per_patient',
              'publication_version_type_4th', 'publication_version_type_5th',
              'disease_free_status_1:Recurred/Progressed', 'new_neoplasm_post_therapy_YES',
              'tissue_retrospective_collection_YES']

target = 'overall_survival_months'
```

In [64]:

```
X = relevant_df[predictors]
y = relevant_df[target]
```

In [65]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [66]:

```
model = LinearRegression()
```

In [67]:

```
model.fit(X_train, y_train)
```

Out[67]:

LinearRegression

LinearRegression()

In [68]:

```
y_pred = model.predict(X_test)
```

In [69]:

```
mse = mean_squared_error(y_test, y_pred)
r_squared = r2_score(y_test, y_pred)
```

In [70]:

```
print("Mean Squared Error:", mse)
print("R-squared:", r_squared)
print("Model Coefficients:")
for feature, coef in zip(predictors, model.coef_):
    print(f"{feature}: {coef}")
```

```
Mean Squared Error: 0.26115967853605937
R-squared: 0.8026098601335447
Model Coefficients:
days_to_sample_collection: 0.17182233107703895
disease_free_months: 0.5670724399228575
samples_per_patient: -0.19952033699331378
publication_version_type_4th: 0.3971847555306206
publication_version_type_5th: 0.49110314440735836
disease_free_status_1:Recurred/Progressed: 0.7436510676758373
new_neoplasm_post_therapy_YES: 0.18444962600957643
tissue_retrospective_collection_YES: 0.02265782572561742
```


In [71]:

```
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
```

In [72]:

```
svr_model = SVR()
dtr_model = DecisionTreeRegressor(random_state=42)
rfr_model = RandomForestRegressor(random_state=42)
gbr_model = GradientBoostingRegressor(random_state=42)
```

In [73]:

```
svr_model.fit(X_train, y_train)
```

Out[73]:

SVR

SVR()

In [74]:

```
dtr_model.fit(X_train, y_train)
```

Out[74]:

```
DecisionTreeRegressor
DecisionTreeRegressor(random_state=42)
```

In [75]:

```
rfr_model.fit(X_train, y_train)
```

Out[75]:

```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

In [76]:

```
gbr_model.fit(X_train, y_train)
```

Out[76]:

```
GradientBoostingRegressor
GradientBoostingRegressor(random_state=42)
```

In [77]:

```
svr_predictions = svr_model.predict(X_test)
dtr_predictions = dtr_model.predict(X_test)
rfr_predictions = rfr_model.predict(X_test)
gbr_predictions = gbr_model.predict(X_test)
```

In [78]:

```
svr_mse = mean_squared_error(y_test, svr_predictions)
svr_r_squared = r2_score(y_test, svr_predictions)

dtr_mse = mean_squared_error(y_test, dtr_predictions)
dtr_r_squared = r2_score(y_test, dtr_predictions)

rfr_mse = mean_squared_error(y_test, rfr_predictions)
rfr_r_squared = r2_score(y_test, rfr_predictions)

gbr_mse = mean_squared_error(y_test, gbr_predictions)
gbr_r_squared = r2_score(y_test, gbr_predictions)

print("SVR - Mean Squared Error:", svr_mse)
print("SVR - R-squared:", svr_r_squared)

print("Decision Tree - Mean Squared Error:", dtr_mse)
print("Decision Tree - R-squared:", dtr_r_squared)

print("Random Forest - Mean Squared Error:", rfr_mse)
print("Random Forest - R-squared:", rfr_r_squared)

print("Gradient Boosting - Mean Squared Error:", gbr_mse)
print("Gradient Boosting - R-squared:", gbr_r_squared)
```

```
SVR - Mean Squared Error: 0.34192438118963875
SVR - R-squared: 0.7415661490889185
Decision Tree - Mean Squared Error: 0.921625337479472
Decision Tree - R-squared: 0.3034156141970301
Random Forest - Mean Squared Error: 0.2811744082890948
Random Forest - R-squared: 0.7874822940119794
Gradient Boosting - Mean Squared Error: 0.2836711344400399
Gradient Boosting - R-squared: 0.7855952143260729
```

In [79]:

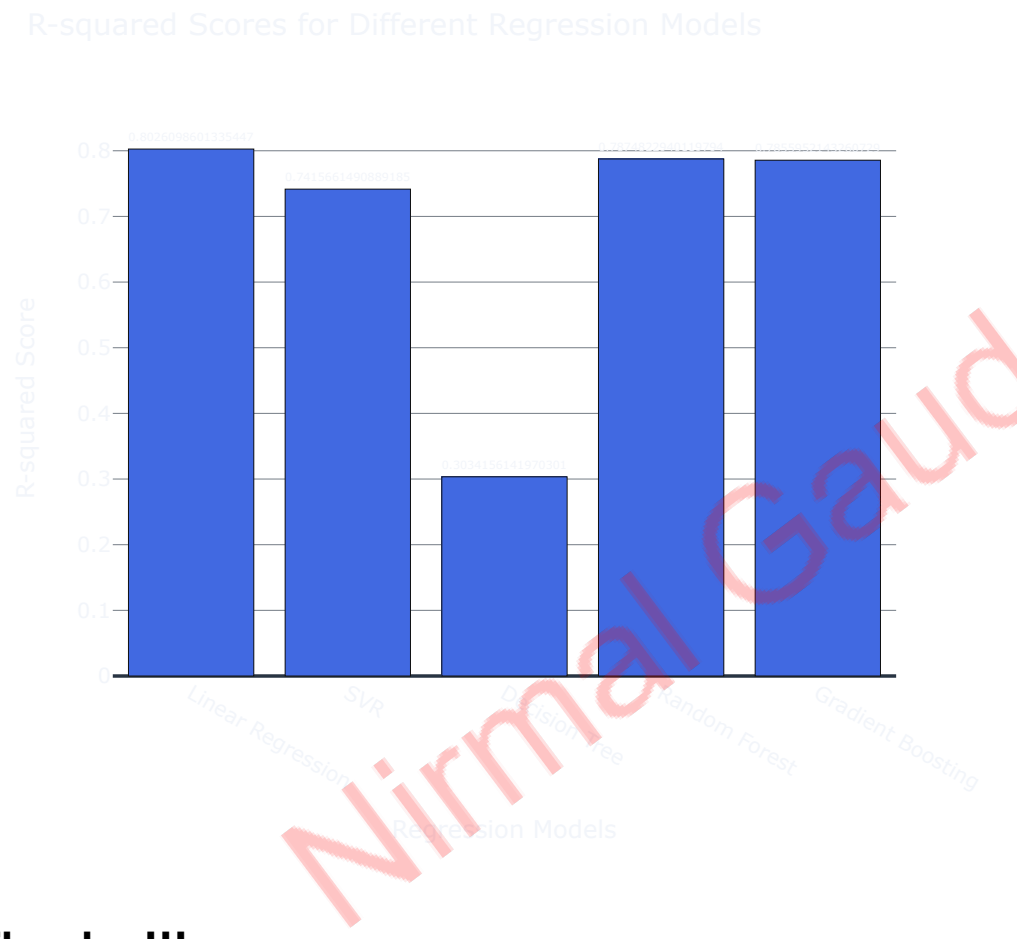
```
r_squared_scores = [r_squared, svr_r_squared, dtr_r_squared, rfr_r_squared, gbr_r_squared]
models = ['Linear Regression', 'SVR', 'Decision Tree', 'Random Forest', 'Gradient Boosting']
```

In [80]:

```
fig = go.Figure([go.Bar(x=models, y=r_squared_scores, text=r_squared_scores,
                        textposition='outside', marker_color='royalblue')])

fig.update_layout(title='R-squared Scores for Different Regression Models',
                  xaxis_title='Regression Models',
                  yaxis_title='R-squared Score',
                  template='plotly_dark')

fig.show()
```



Thanks !!!