In [3]:
```
import numpy as np
import pandas as pd
```

In [4]:
```
df = pd.read_csv("fetal_health.csv")
df.head()
```

Out[4]:

	baseline value	accelerations	fetal_movement	uterine_contractions	light_decelerations	severe_decelerations	prolongued_decelerations	abnormal_short_term_variability	mean_value_of_short_term_variability
0	120.0	0.000	0.0	0.000	0.000	0.0	0.0	73.0	0.5
1	132.0	0.006	0.0	0.006	0.003	0.0	0.0	17.0	2.1
2	133.0	0.003	0.0	0.008	0.003	0.0	0.0	16.0	2.1
3	134.0	0.003	0.0	0.008	0.003	0.0	0.0	16.0	2.4
4	132.0	0.007	0.0	0.008	0.000	0.0	0.0	16.0	2.4

5 rows × 22 columns

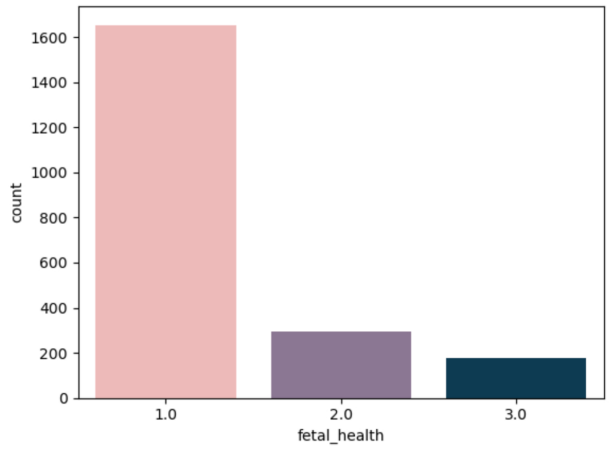
In [5]:
```
df.describe().T
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
baseline value	2126.0	133.303857	9.840844	106.0	126.000	133.000	140.000	160.000
accelerations	2126.0	0.003178	0.003866	0.0	0.000	0.002	0.006	0.019
fetal_movement	2126.0	0.009481	0.046666	0.0	0.000	0.000	0.003	0.481
uterine_contractions	2126.0	0.004366	0.002946	0.0	0.002	0.004	0.007	0.015
light_decelerations	2126.0	0.001889	0.002960	0.0	0.000	0.000	0.003	0.015
severe_decelerations	2126.0	0.000003	0.000057	0.0	0.000	0.000	0.000	0.001
prolongued_decelerations	2126.0	0.000159	0.000590	0.0	0.000	0.000	0.000	0.005
abnormal_short_term_variability	2126.0	46.990122	17.192814	12.0	32.000	49.000	61.000	87.000
mean_value_of_short_term_variability	2126.0	1.332785	0.883241	0.2	0.700	1.200	1.700	7.000
percentage_of_time_with_abnormal_long_term_variability	2126.0	9.846660	18.396880	0.0	0.000	0.000	11.000	91.000
mean_value_of_long_term_variability	2126.0	8.187629	5.628247	0.0	4.600	7.400	10.800	50.700
histogram_width	2126.0	70.445908	38.955693	3.0	37.000	67.500	100.000	180.000
histogram_min	2126.0	93.579492	29.560212	50.0	67.000	93.000	120.000	159.000
histogram_max	2126.0	164.025400	17.944183	122.0	152.000	162.000	174.000	238.000
histogram_number_of_peaks	2126.0	4.068203	2.949386	0.0	2.000	3.000	6.000	18.000
histogram_number_of_zeroes	2126.0	0.323612	0.706059	0.0	0.000	0.000	0.000	10.000
histogram_mode	2126.0	137.452023	16.381289	60.0	129.000	139.000	148.000	187.000
histogram_mean	2126.0	134.610536	15.593596	73.0	125.000	136.000	145.000	182.000
histogram_median	2126.0	138.090310	14.466589	77.0	129.000	139.000	148.000	186.000
histogram_variance	2126.0	18.808090	28.977636	0.0	2.000	7.000	24.000	269.000
histogram_tendency	2126.0	0.320320	0.610829	-1.0	0.000	0.000	1.000	1.000
fetal_health	2126.0	1.304327	0.614377	1.0	1.000	1.000	1.000	3.000

In [6]:
```
import seaborn as sns
colours=["#f7b2b0","#8f7198", "#003f5c"]
sns.countplot(data= df, x="fetal_health",palette=colours)
```

Out[6]:
```
<Axes: xlabel='fetal_health', ylabel='count'>
```

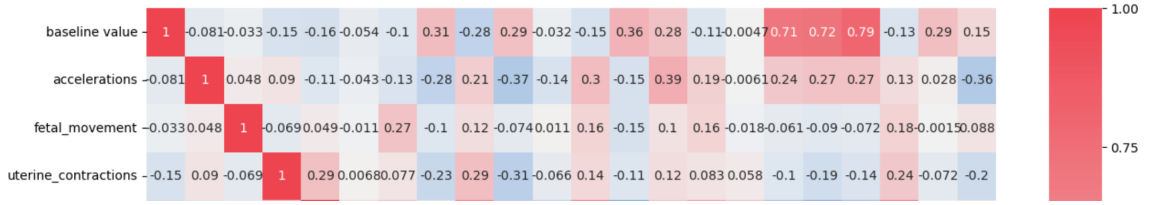


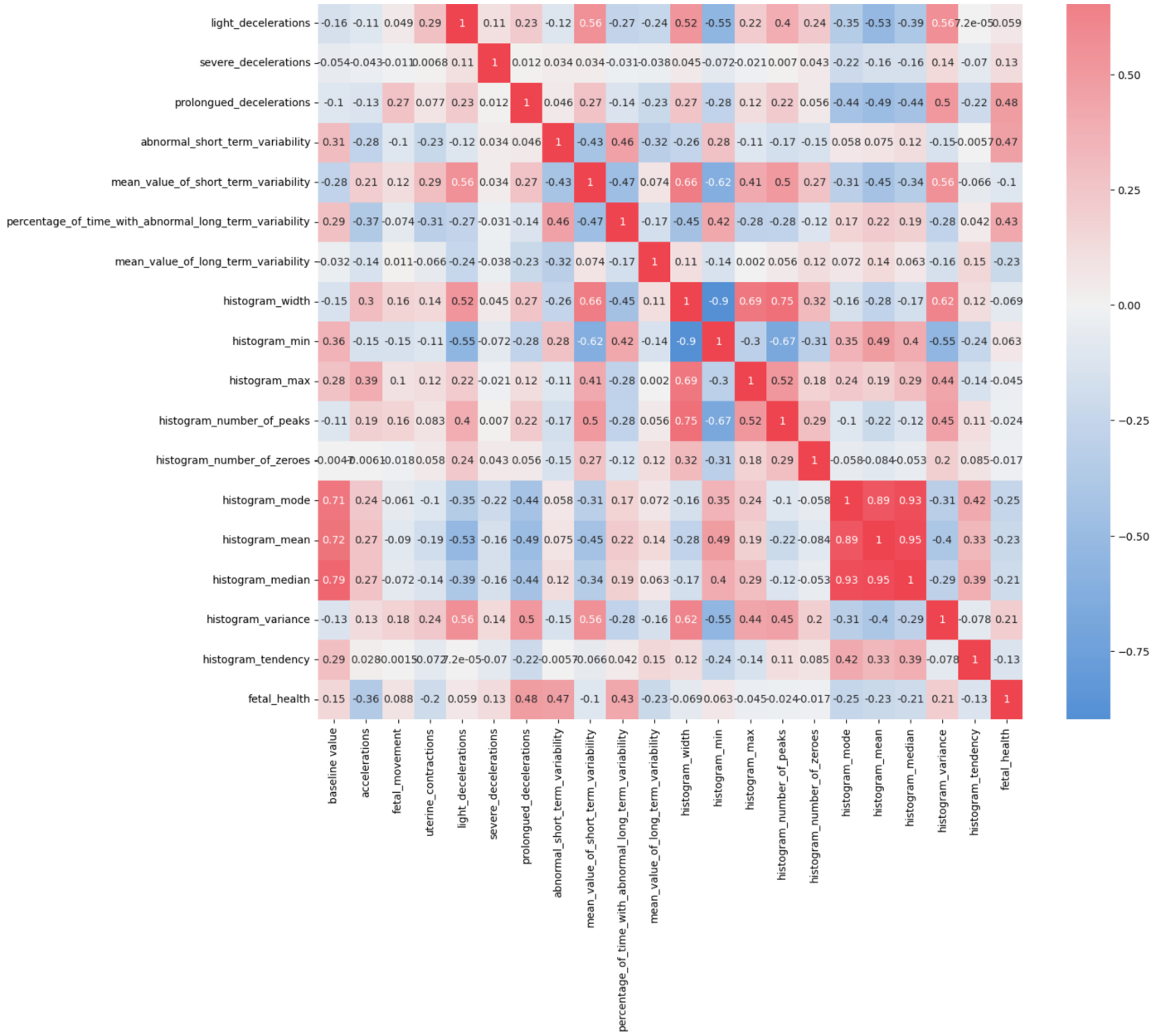
In [7]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(15,15))

cmap = sns.diverging_palette(250, 10, s=80, l=55, n=9, as_cmap=True)

sns.heatmap(df.corr(),annot=True, cmap=cmap, center=0)
```

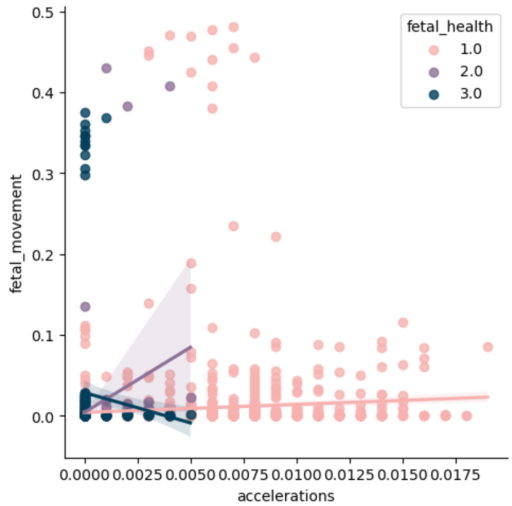
Out[7]:
```
<Axes: >
```





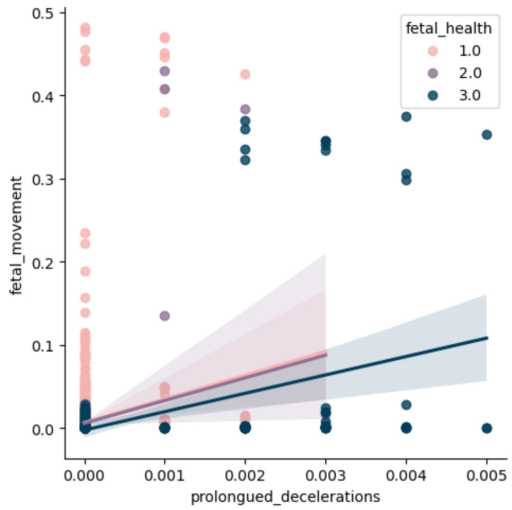
In [8]:
```
sns.lmplot(data =df,x="accelerations",y="fetal_movement",palette=colours, hue="fetal_health",legend_out=False)
plt.show()
```

```
/Users/vinay/anaconda3/lib/python3.10/site-packages/seaborn/regression.py:582: UserWarning: legend_out is deprecated from the `lmplot` function signature. Please update
your code to pass it using `facet_kws`.
  warnings.warn(msg, UserWarning)
```



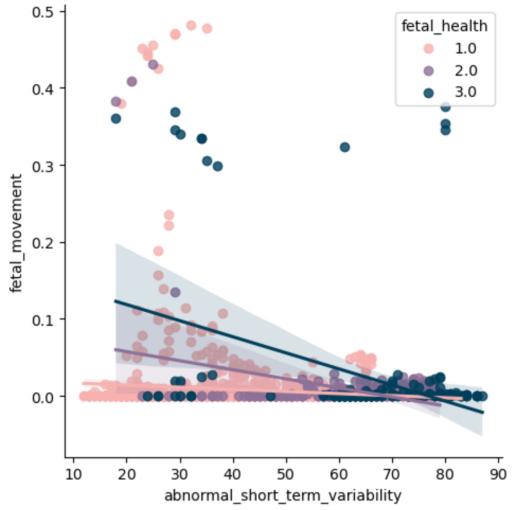
In [9]:
```
sns.lmplot(data =df,x="prolongued_decelerations",y="fetal_movement",palette=colours, hue="fetal_health",legend_out=False)
plt.show()
```

```
/Users/vinay/anaconda3/lib/python3.10/site-packages/seaborn/regression.py:582: UserWarning: legend_out is deprecated from the `lmplot` function signature. Please update
your code to pass it using `facet_kws`.
```



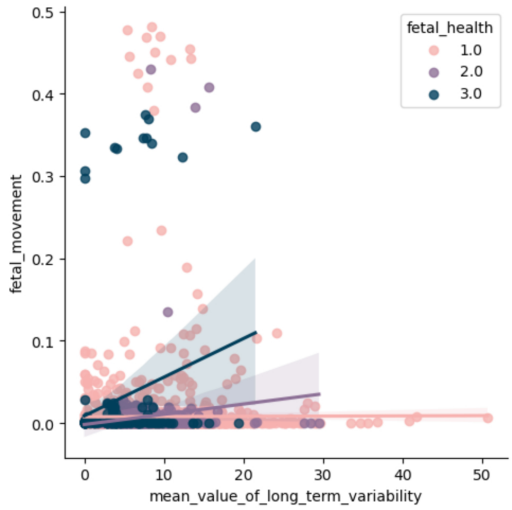
In [10]:
```
sns.lmplot(data =df,x="abnormal_short_term_variability",y="fetal_movement",palette=colours, hue="fetal_health",legend_out=False)
plt.show()
```

```
/Users/vinay/anaconda3/lib/python3.10/site-packages/seaborn/regression.py:582: UserWarning: legend_out is deprecated from the `lmplot` function signature. Please update
your code to pass it using `facet_kws`.
  warnings.warn(msg, UserWarning)
```

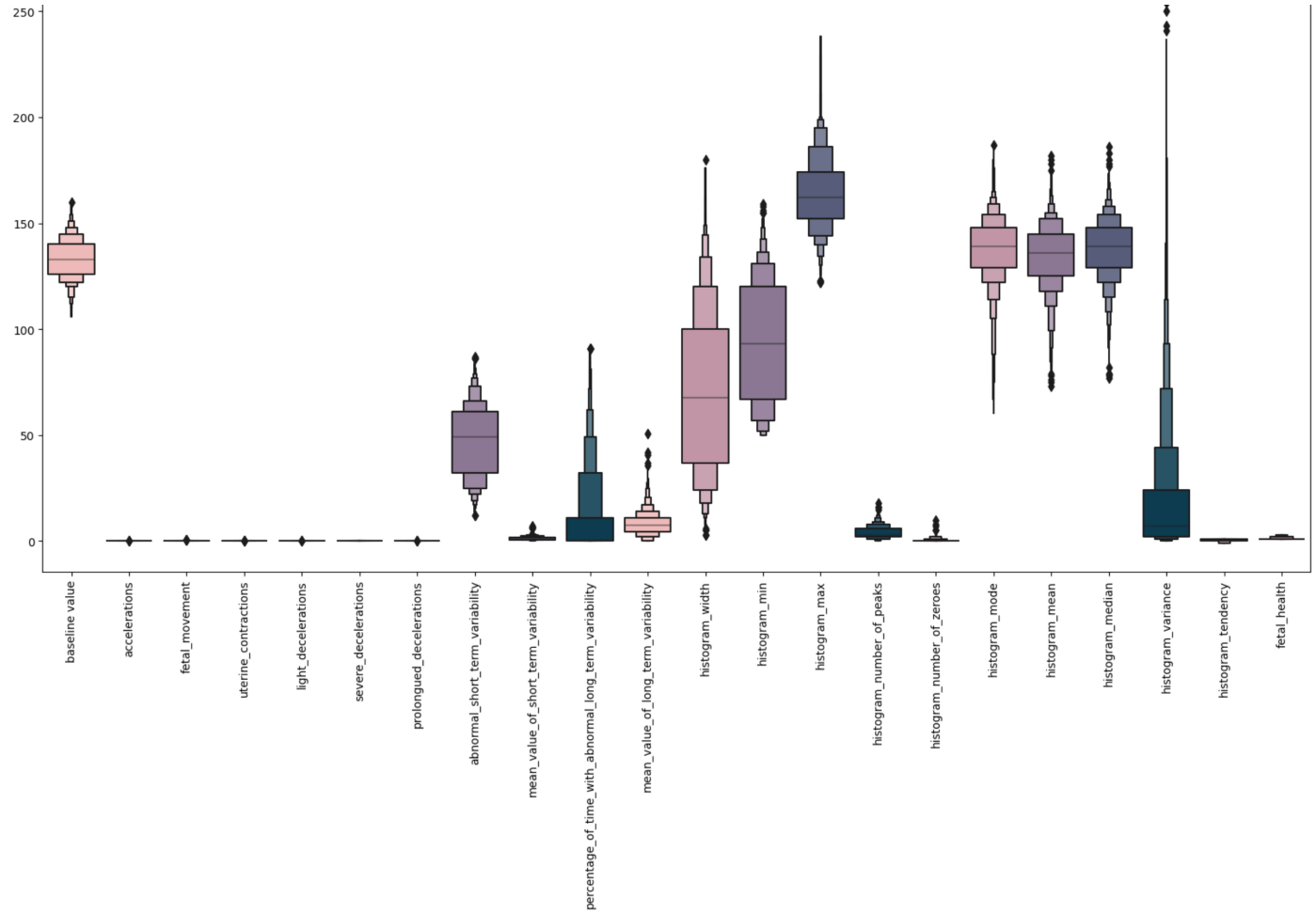


In [11]:
```
sns.lmplot(data =df,x="mean_value_of_long_term_variability",y="fetal_movement",palette=colours, hue="fetal_health",legend_out=False)
plt.show()
```

```
/Users/vinay/anaconda3/lib/python3.10/site-packages/seaborn/regression.py:582: UserWarning: legend_out is deprecated from the `lmplot` function signature. Please update
your code to pass it using `facet_kws`.
  warnings.warn(msg, UserWarning)
```



In [10]:
```
shades =["#f7b2b0","#c98ea6","#8f7198","#50587f", "#003f5c"]
plt.figure(figsize=(20,10))
sns.boxenplot(data = df,palette = shades)
plt.xticks(rotation=90)
plt.show()
```



In [11]:
```
X=df.drop(["fetal_health"],axis=1)
y=df["fetal_health"]
```

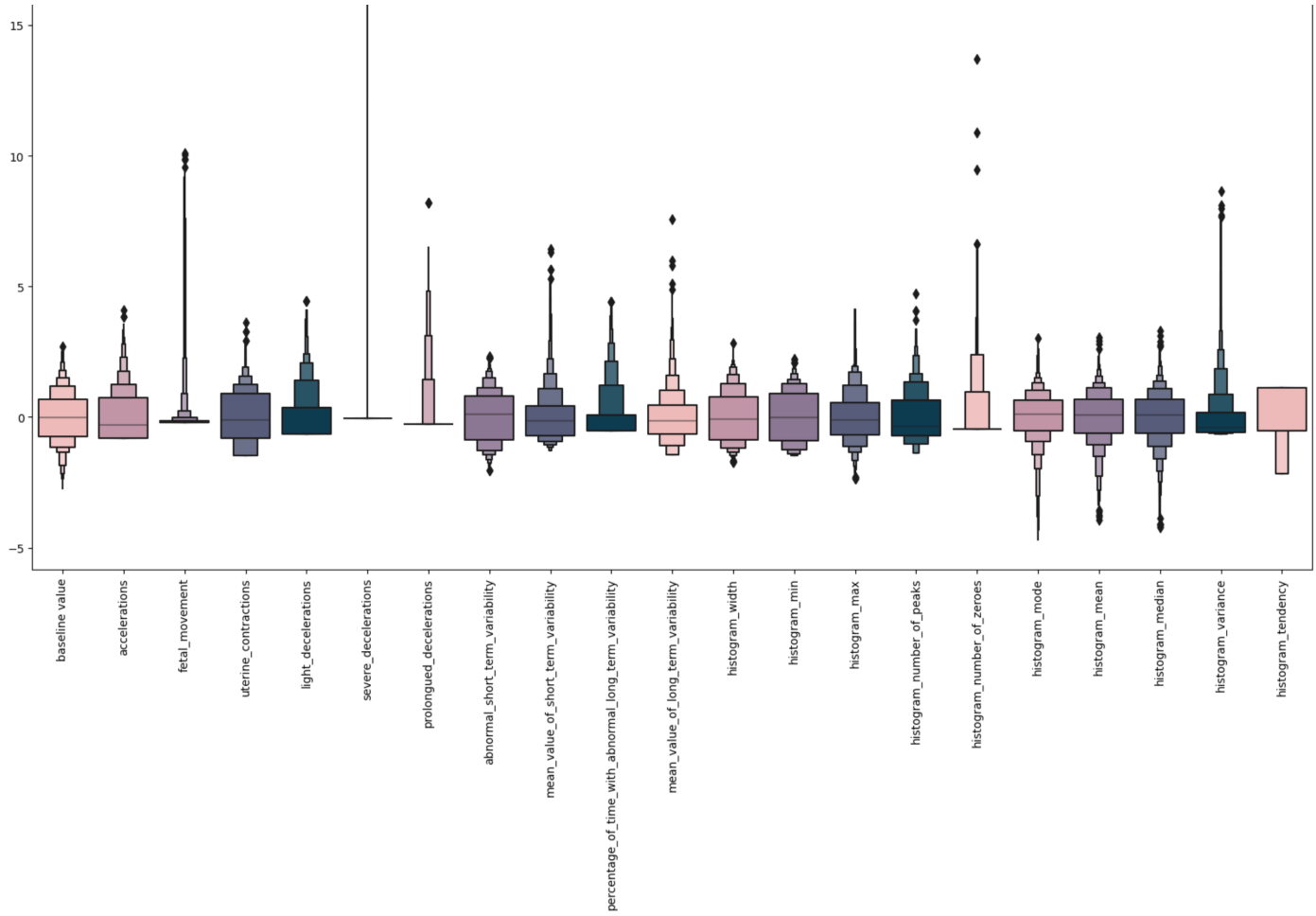
In [12]:
```
from sklearn.preprocessing import StandardScaler
col_names = list(X.columns)
```

In [13]:
```
from sklearn import preprocessing
scaler = preprocessing.StandardScaler()
X_df= scaler.fit_transform(X)
X_df = pd.DataFrame(X_df, columns=col_names)
X_df.describe().T
```

Out[13]:

	count	mean	std	min	25%	50%	75%	max
baseline value	2126.0	1.069490e-15	1.000235	-2.775197	-0.742373	-0.030884	0.680604	2.713428
accelerations	2126.0	-4.010589e-17	1.000235	-0.822388	-0.822388	-0.304881	0.730133	4.093929
fetal_movement	2126.0	-1.336863e-17	1.000235	-0.203210	-0.203210	-0.203210	-0.138908	10.106540
uterine_contractions	2126.0	-1.336863e-16	1.000235	-1.482465	-0.803434	-0.124404	0.894142	3.610264
light_decelerations	2126.0	-5.347452e-17	1.000235	-0.638438	-0.638438	-0.638438	0.375243	4.429965
severe_decelerations	2126.0	6.684315e-18	1.000235	-0.057476	-0.057476	-0.057476	-0.057476	17.398686
prolongued_decelerations	2126.0	1.336863e-17	1.000235	-0.268754	-0.268754	-0.268754	-0.268754	8.208570
abnormal_short_term_variability	2126.0	-7.352747e-17	1.000235	-2.035639	-0.872088	0.116930	0.815060	2.327675
mean_value_of_short_term_variability	2126.0	6.684315e-17	1.000235	-1.282833	-0.716603	-0.150373	0.415857	6.417893
percentage_of_time_with_abnormal_long_term_variability	2126.0	-5.347452e-17	1.000235	-0.535361	-0.535361	-0.535361	0.062707	4.412293
mean_value_of_long_term_variability	2126.0	2.406354e-16	1.000235	-1.455081	-0.637583	-0.139975	0.464263	7.555172
histogram_width	2126.0	-3.007942e-17	1.000235	-1.731757	-0.858765	-0.075640	0.758838	2.812936
histogram_min	2126.0	-4.679021e-17	1.000235	-1.474609	-0.899376	-0.019608	0.893996	2.213648
histogram_max	2126.0	-1.203177e-16	1.000235	-2.342558	-0.670314	-0.112899	0.555999	4.123453
histogram_number_of_peaks	2126.0	-1.671079e-16	1.000235	-1.379664	-0.701397	-0.362263	0.655137	4.724738
histogram_number_of_zeroes	2126.0	2.757280e-17	1.000235	-0.458444	-0.458444	-0.458444	-0.458444	13.708003
histogram_mode	2126.0	1.069490e-16	1.000235	-4.729191	-0.516077	0.094519	0.644055	3.025381
histogram_mean	2126.0	-6.684315e-16	1.000235	-3.951945	-0.616458	0.089126	0.666422	3.039749
histogram_median	2126.0	2.673726e-16	1.000235	-4.223849	-0.628514	0.062897	0.685166	3.312527
histogram_variance	2126.0	-5.347452e-17	1.000235	-0.649208	-0.580173	-0.407586	0.179212	8.635997
histogram_tendency	2126.0	-1.069490e-16	1.000235	-2.162031	-0.524526	-0.524526	1.112980	1.112980

In [14]:
```
plt.figure(figsize=(20,10))
sns.boxenplot(data = X_df,palette = shades)
plt.xticks(rotation=90)
plt.show()
```



In [15]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train,y_test = train_test_split(X_df,y,test_size=0.3,random_state=42)
```

In [16]:
```
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
```

Out[16]:
```
▾ RandomForestClassifier
RandomForestClassifier()
```

In [17]:
```
from sklearn.metrics import precision_score, recall_score, confusion_matrix, classification_report, accuracy_score, f1_score
pred_rf = rf.predict(X_test)
accuracy = accuracy_score(y_test, pred_rf)
print(accuracy)
```

```
0.9404388714733543
```

In [18]:
```
from sklearn.model_selection import GridSearchCV
from sklearn.model_selection import cross_val_score
parameters = {
    'n_estimators': [100,150, 200,500,700,900],
    'max_features': ['auto', 'sqrt', 'log2'],
    'max_depth' : [4,6,8,12,14,16],
    'criterion' :['gini', 'entropy'],
    'n_jobs':[-1,1,None]
}

#Fitting the trainingset to find parameters with best accuracy

CV_rfc = GridSearchCV(estimator=RandomForestClassifier(), param_grid=parameters, cv= 5)
```

In [19]:
```
RF_model = RandomForestClassifier(criterion= 'entropy',
 max_depth= 12,
 max_features= 'auto',
 n_estimators= 150,
 n_jobs= None)
RF_model.fit(X_train, y_train)
#Testing the Model on test set
predictions=RF_model.predict(X_test)
acccuracy= accuracy_score(y_test,predictions)
acccuracy
```

```
/Users/vinay/anaconda3/lib/python3.10/site-packages/sklearn/ensemble/_forest.py:424: FutureWarning: `max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To keep the past behaviour, explicitly set `max_features='sqrt'` or remove this parameter as it is also the default value for RandomForestClassifiers and ExtraTreesClassifiers.
  warn(
```

Out[19]:
```
0.945141065830721
```

In [20]:
```
acccuracy = accuracy_score(y_test, predictions)
recall = recall_score(y_test, predictions, average="weighted")
precision = precision_score(y_test, predictions, average="weighted")
f1_score = f1_score(y_test, predictions, average="micro")

print("********* Random Forest Results *********")
print("Accuracy    : ", acccuracy)
print("Recall      : ", recall)
```

```
print("Precision   : ", precision)
print("F1 Score    : ", f1_score)
```

```
********* Random Forest Results *********
Accuracy    :  0.945141065830721
Recall      :  0.945141065830721
Precision   :  0.9441992090178781
F1 Score    :  0.945141065830721
```

In [21]:
```
rf2 = RandomForestClassifier(criterion="gini", n_estimators = 100, min_samples_leaf=1, min_samples_split=2, random_state=42)
rf2.fit(X_train, y_train)
predictions2=rf2.predict(X_test)
acccuracy= accuracy_score(y_test,predictions2)
```

In [22]:
```
acccuracy
```

Out[22]:
```
0.9435736677115988
```

In [23]:
```
confusion_matrix(y_test, predictions)
```

Out[23]:
```
array([[487,   7,   2],
       [ 20,  78,   3],
       [  2,   1,  38]])
```

In [24]:
```
confusion_matrix(y_test, predictions2)
```

Out[24]:
```
array([[487,   7,   2],
       [ 21,  77,   3],
       [  2,   1,  38]])
```