

```

// The following code is to be added to OverrideViewModel.cs for additional
functionality

// This part goes to the "if" statement in "public string StructureSelect{}" to
add a messagebox for the user to choose yes or no.
        DialogResult dialogResult = MessageBox.Show("Do you want to
calculate imaging statistics for this structure?", "", MessageBoxButtons.YesNo);
        if (dialogResult == DialogResult.Yes)
        {
            AnalyzeStructureVoxels();
        }
        else
        {
            TextBox = "Please enter the intended CT number for this
structure and click the Convert button to start image conversion.";
        }

// This is a method to iterate voxels inside a structure set and do some
analysis.
    private void AnalyzeStructureVoxels()
    {
        _worker.Run(scriptContext =>
        {
            TextBox = $"Reading image data...";
            ImageSelectionEnabled = false;
            StructureSelectionEnabled = false;
            IsOverrideBtnEnabled = false;
            var seriesId =
ImageSelected.Split('(')[0].Remove(ImageSelected.Split('(')[0].Length - 1);
            var imageId = ImageSelected.Split('(')[1].Split(' ')[0];
            var CurrentImage3D =
scriptContext.Patient.Studies.SelectMany(study =>
study.Images3D).ToList().Where(img =>
            (img.Series.Id == seriesId && img.Id ==
imageId)).FirstOrDefault();
            var structureset = scriptContext.Patient.StructureSets.Where(s =>
(s.Image.Id == imageId && s.Image.Series.Id == seriesId)).FirstOrDefault();
            Structure structure = structureset.Structures.Where(s => s.Id ==
StructureSelected).FirstOrDefault();
            int ZSlices = CurrentImage3D.ZSize;
            System.Collections.BitArray segmentStride = new
System.Collections.BitArray((int)CurrentImage3D.XSize);
            double[] imagePixels = new double[((int)CurrentImage3D.XSize)];
            List<double> numbers = new List<double>();

```

```

        for (int Z = 0; Z < ZSlices; Z++)
        {
            int[,] voxelPlane = new int[CurrentImage3D.XSize,
CurrentImage3D.YSize];
            CurrentImage3D.GetVoxels(Z, voxelPlane);
            for (int Y = 0; Y < CurrentImage3D.YSize; Y++)
            {
                var start = CurrentImage3D.Origin +
CurrentImage3D.YDirection * Y * CurrentImage3D.YRes + CurrentImage3D.ZDirection *
Z * CurrentImage3D.ZRes;
                var end = start + CurrentImage3D.XDirection *
CurrentImage3D.XRes * CurrentImage3D.XSize;
                var structProfile = structure.GetSegmentProfile(start,
end, segmentStride);
                CurrentImage3D.GetImageProfile(start, end, imagePixels);
                for (int X = 0; X < CurrentImage3D.XSize; X++)
                {
                    if (segmentStride[X])
                    {
                        numbers.Add(imagePixels[X]);
                    }
                }
            }
            int percent = (int)((float)(Z + 1) / ZSlices * 100);
            ProgressBarValue = percent;
        }
        int count = numbers.Count;
        double avg = numbers.Average();
        double sum = numbers.Sum(d => (d - avg) * (d - avg));
        double stddev = Math.Sqrt(sum / count);
        TextBox = $"This structure includes {numbers.Count} voxels.\n";
        TextBox += $"Average CT number for this structure:
{string.Format("{0:0.0} HU", avg)} with StdDev: {string.Format("{0:0.0}",
stddev)}\n\n";
        TextBox += "Next, please enter the intended CT number for this
structure and click the Convert button to start image conversion.";
        ImageSelectionEnabled = true;
        StructureSelectionEnabled = true;
        IsOverrideBtnEnabled = true;
    });
}

// This is a method for synchronous ESAPI work.
public void RunSynchronously(Action<ScriptContext> a)
{

```

```
        // The Invoke method executes the delegate synchronously.  
        _dispatcher.Invoke(a, _scriptContext);  
    }
```