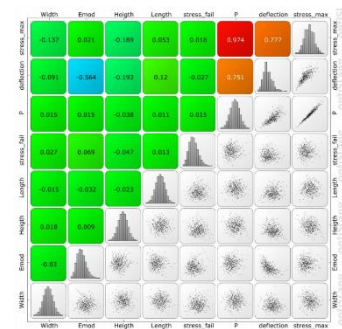
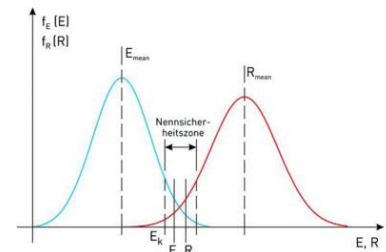


# Final Project: Visualization

By Paul Debus, Paul Brust

## Grundidee:

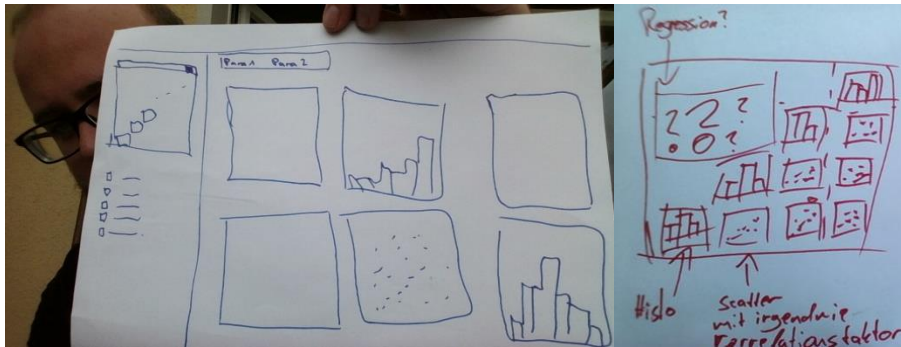
Im Konstruktiven Bauingenieurwesen geht es i.d.R. darum Standsicherheitsnachweise für alle Gebäudeteile durchzuführen. Traditionell wird dabei in den geltenden Europäischen Normen auf ein semi-probabilistisches Sicherheitskonzept zurück gegriffen. Einen moderneren Ansatz zur besseren Materialausnutzung bietet heutzutage allerdings die Simulation beispielsweise via Finite Element Modelling. Durch diese können beispielsweise auch Stochastische Berechnungen durchgeführt werden. Dabei werden Samples mit verschiedener Parameterstreuung erzeugt, die Modelle damit berechnet und aus den Ergebnissen Rückschlüsse auf die Versagenswahrscheinlichkeit der Struktur geschlossen. Im Zuge eines anderen Kurses haben wir vor kurzen mit dem Programm Optislang der Firma Dynardo gearbeitet, das solche Daten visualisiert um Rückschlüsse des Tragwerksverhaltens in Abhängigkeit einzelner Eingabeparameter und deren Sensitivität zu ermöglichen. Während dieses Tool bereits einige Visualisierungstechniken verwendet und auch gut umsetzen kann, haben wir feststellen müssen, dass sowohl in der Benutzerfreundlichkeit, als auch bei den verwendeten Techniken unserer Meinung nach Verbesserungspotential existiert. Als Final Projekt haben wir daher mit selbst generierten stochastischen Daten für ein statisches Problem in Anlehnung an einen Teil des besagten Programms eine eigene Visualisierung umgesetzt um unsere Ideen auszuprobieren.



Unser Anfangsentwurf sah eine horizontale Zweiteilung des Bildschirms vor, bei dem der linke Teil ein Auswahlmenü der zu zeigenden Daten darstellt und der rechte Teil den Nutzer über das jeweils ausgewählte Datenpaar informiert.

Die Datenauswahl sollte über eine Scatterplotmatrix ähnlich der Dynardos geschehen, die Correlationsfaktoren allerdings nicht durch farbige Felder in der oberen Hälfte, sondern bereits mit den darstellenden Scatterplots in der unteren Hälfte erfolgen, um eine leichtere Zuordnung zu ermöglichen. Der Plan sah vor die obere Hälfte zur Darstellung weiterer Informationen zu nutzen und eine visuelle Trennung von Eingabe und Ausgabeparametern zu erzeugen. Welche Parameter in der Matrix angezeigt werden sollte über eine Checkboxliste geregelt sein, ein Doppelklick auf ein Scatterplot sollte dieses hervorholen und Interaktionen wie Zoom, brushing and linking erlauben.

Die rechte Seite des Bildschirms sollte ein Feld sein, auf dem einzelne Subwindows erzeugt werden können. Diese sollten, je nach dem welches Feld der Scatterplotmatrix ausgewählt wurde, verschiedene Darstellung der dadurch ausgewählten Datenpaare zeigen. Angedacht wurden vergrößerte Histogramme und Scatterplots, Jitterplots, sowie die Anzeige von statistischen Daten und genaue Informationen über vom Nutzer ausgewählte Punkte. Ein Vergrößern des jeweiligen Fensters auf volle Bildschirmgröße sollte ggf. weitere Informationen zu den jeweiligen Darstellungen bieten.



## Umsetzung:

Unser Programm liest eine CSV-Datei ein und speichert die darin enthaltenen Daten in einem Model für unsere Darstellung ein. Es beinhaltet eine Liste der Input-, sowie der Outputvariablen, einer Liste aller Datenpunkte für statistische Werte der einzelnen Verteilungen und einer Korrelationsmatrix, die direkt zu Beginn aus den Daten errechnet werden, sowie mehrere Listen für die Steuerung der vom Nutzer zur Anzeige ausgewählten Datensätze und Datenpunkte.

Zur Nutzerinteraktion wurde ein GUI mit Hilfe des Qt-Designers erzeugt. Dieses wird via PyQt von dem von uns geschriebenen Python-Code verwendet. Die geplante Bildschirmaufteilung wurde durch einen Splitter implementiert und bleibt dadurch dynamisch Verschiebbar, um dem Nutzer eine Möglichkeit zum Vergrößern und Verkleinern des jeweils wichtigen Bildschirmteils zu ermöglichen. Auf der linken Seite wurde wie geplant eine Scatterplotmatrix und ein Feld mit Checkboxes eingefügt um eine Datenauswahl zu ermöglichen. Die Liste der Checkboxes wird anhand der im Model vorhandenen Liste von Input- und Output Parametern generiert, wobei die Inputvariablen grau hinterlegt sind. Zu Beginn sind nur die maximal ersten sieben Parameter ausgewählt und angezeigt um den Benutzer nicht schon am Anfang mit zu vielen Informationen zu überfordern. Die geplante Darstellung des Korrelationsfaktors wurde einfach durch eine Färbung des Hintergrundes der Scatterplots realisiert, dies gewährleistet ein schnelles identifizieren von korrelierten Datensätzen auch bei sehr stark verkleinerten Scatterplots und großer Matrizen. Die Farbpalette wurde so gewählt, dass diese, für den Ingenieur wichtigsten Datensätze, deutlich erkennbar und hervorgehoben sind. Im oberen freien Bereich wurde lediglich ein Colorbar der verwendeten Palette eingefügt um das Interface überschaubar zu halten. Von der Idee einzelne Scatterplots durch Doppelklick in der Matrix zu vergrößern und hier brushing and linking zu ermöglichen wurde während der Implementierung Abstand genommen, da diese Funktion im rechten Bereich durch die Subwindows ermöglicht wird.

Auf der rechten Seite werden für das ausgewählte Datenpaar immer die Histogramme beider Parameter, der Scatterplot beider, sowie ein Fenster mit in Text gefassten statistischen Informationen erzeugt. Zusätzlich können drei weitere Darstellungsformen erzeugt werden: Ein Starplot das die Korrelation eines Parameters mit allen Parametern darstellt, ein Pieplot das den Anteil einzelner Inputparameter an der Gesamtvarianz eines Outputparameters anzeigt, sowie eine Art Jitterplot zur Darstellung der statistischen Eigenschaften der Datenpunkte entlang einer Parameterachse. Diese werden per Default nur angezeigt, wenn das ausgewählte Datenpaar bestimmte Treshholds übersteigt die die Darstellungen sinnvoll erscheinen lassen. Wie alle Plots können sie allerdings jederzeit durch einen Menüreiter im oberen Bildschirmbereich zusätzlich erzeugt werden. Wenn ein Histogramm auf Vollbild vergrößert wird, wird zusätzlich ein Curve-fitting durchgeführt. Im unteren Bereich erscheint ein zusätzliches Label, dass den wahrscheinlichsten Verteilungstyp sowie dessen Eigenschaften anzeigt. Bereits im kleinen Scatterplot ist verschieben und zoomen möglich. Auch die Auswahl von Datenpaaren durch ein Auswahlrechteck, das durch Drücken und Halten der rechten Maustaste erzeugt wird, wird bereits im kleinen Fenster unterstützt. Die ausgewählten Punkte werden, solange nicht neu markiert wird, in allen Scatterplots die im rechten

Fenster erzeugt werden farblich markiert und hervorgehoben dargestellt. Zusätzlich erscheint eine Tabelle mit allen Werten der ausgewählten Punkte im Textfenster. Eine Vergrößerung der Scatterplotmatrix auf Vollbild schaltet zusätzlich eine Overviewansicht in der linken oberen Ecke frei, um die Orientierung auch bei Zoom und Verschieben der Daten zu gewährleisten.

Das im Reiter als „Mean-Std“-Plot bezeichnete Diagramm soll die Änderung der statistischen Eigenschaften einer Verteilung entlang eines Parameters von stärker korrelierten Datensätzen darzustellen. Ähnlich Jitterplots werden die einzelnen Daten in bestimmten Intervallen zusammengefasst. Anstatt ihre Anzahl darzustellen wird jedoch für jedes Intervall Mittel-, Maximal- und Minimalwert sowie die Standardabweichung ermittelt und diese als einzelne Linien und Flächen dargestellt. Starplots, die automatisch erzeugt werden, wenn ein ausgewählter Parameter starke Korrelation mit mehr als einem weiteren Parameter aufweist, zeigen den Korrelationsfaktor eines Parameters mit allen Parametern an. Sie erlauben eine schnelle Identifikation von Einflüssen der Parameter zueinander. Die letzte Art Diagramm welches nur für Outputvariablen anzeigbar ist, stellt das „Var-Pie“ Diagramm dar. Hierfür wird der „Coefficient of Correlation“ ähnlich dem von Dynardos Optislang verwendeten Modells berechnet. Aber anders als dort werden die Ergebnisse nicht als Bar-Graph sondern als Torte dargestellt, da der für jeden Parameter ermittelte Wert einem prozentualen Anteil an der Gesamtvarianz darstellt. Um erhaltene Erkenntnisse speichern zu können wurde zudem noch eine Exportfunktion hinzugefügt, die das aktuell ausgewählte Subwindow als JPG in einem Ordner innerhalb des Programms abspeichert.

