

Case Study im Modul Applied Data Science for Quality Engineering

Ansprechpartner:

Tim-Gunnar Hensel

David Barkemeyer



Ausgabe:

23.05.2019

Inhaltsverzeichnis

1	Einleitung	3
2	Optimierung des DAP-Copter	3
2.1	Aufnahmebogen	4
2.2	Ortsauswahl	4
2.3	Standardisierung	5
2.4	Vollfaktorieller Versuchsplan	5
2.5	Steepest Ascent	6
3	Entwicklung der Applikation	7
3.1	Workflow	7
3.2	Dateien	7
3.3	Anforderungen an die App	8
3.4	Erstellen eines neuen Moduls	9
3.5	Weitere Hinweise	9
4	Präsentation der Case-Study	10

1 Einleitung

Sie sind Mitarbeiter der DQE-Data-Science SE, die auf die Optimierung von Prozessen mithilfe der Methoden des Data Science spezialisiert ist. Als Teil eines fünfköpfigen Teams sollen Sie in den kommenden Wochen der DAP-Copter-GmbH, die leistungsfähige Flugobjekte herstellt, beratend zur Seite stehen. Das einstige Aushängeschild der DAP-Copter-GmbH, der DAP-Copter, ist nicht mehr zeitgemäß und findet immer geringeren Absatz. Vom Management bekommen Sie die Aufgabe die Flugzeit des DAP-Copters signifikant zu steigern (Phase 1). Die dabei gewonnenen Erkenntnisse sollen Sie nutzen, um eine Applikation zu entwickeln, die es ermöglicht, die Produktpalette systematisch zu verbessern (Phase 2).

2 Optimierung des DAP-Copter

Für die erste Projektphase gewährt Ihnen das Management ein Budget von 750.000€. Jeder DAP-Copter-Prototyp kostet Sie 20.000€. Die Kosten für den Test eines DAP-Copters betragen 2.000€. Die Toleranzbreite für die Zielgröße Flugdauer beträgt 0,5 s. DAP-Copter können nach einem Flugtest nicht modifiziert werden. Die Mindestabwurfhöhe beträgt 4 m. Achten Sie bitte unbedingt auf Ihre Sicherheit bei den Versuchen!

Das Referenzdesign ist durch eine Flügel- und Körperlänge von 75 mm, einen Einschnitt von 30 mm sowie eine Papierstärke von 100 g/mm² gegeben. Mögliche Faktoren, die die Flugzeit beeinflussen können, entnehmen Sie Tabelle 1.

Tabelle 1: Faktoren

Faktor	Low	High
Flügelänge	60 mm	90 mm
Körperlänge	50 mm	100 mm
Einschnitt	0 mm	60 mm
Papierstärke	80 g/mm ²	120 g/mm ²

Beginnen Sie mit dem Projekt, indem Sie einen Überblick über das Vorhaben erstellen, nachdem Sie die gesamte Aufgabenstellung durchgelesen haben. Legen Sie folgende Verantwortlichkeiten fest:

- **Testingenieur:** Leitet die Durchführung aller Flugtests. Entscheidet als letzte Instanz, ob Flugtests wiederholt werden müssen.
- **Prüfingenieur:** Leitet die Messungen aller Flugtests. Entscheidet als letzte Instanz, ob die Messung(en) gültig sind.
- **Fertigungsingenieur:** Leitet die Fertigung der Prototypen. Entscheidet als letzte Instanz, wie die DAP-Copter gefertigt werden.

- **Datenerfasser:** Verantwortlich für die Dokumentation der Versuche.
- **Kosteningenieur:** Verantwortlich für die Ausgaben der Projektgruppe. Stellt sicher, dass die Ausgaben die Budgetrestriktionen nicht überschreiten. Kalkuliert die erwarteten Kosten für die Durchführung aller Tests.

Führen Sie die komplette erste Projektphase durch, indem Sie die nachfolgenden Aufgaben bearbeiten und ihre Ergebnisse auswerten. Wenden Sie sich dann der zweiten Phase des Projektes zu, indem Sie Ihre erworbenen Kenntnisse dazu verwenden, eine Shiny-App zu programmieren. Die App soll in der Lage sein, die Auswertungen, die Sie in der ersten Projektphase, getätigt haben, für beliebige Daten mit identischer Datenstruktur auszuführen. Arbeiten Sie sich schon während der ersten Projektphase in das Shiny-Framework ein, da es für gewöhnlich einige Zeit benötigt, die Konzepte zu durchdringen.

Bearbeiten Sie die nachfolgenden Aufgaben der Reihe nach und dokumentieren Sie Ihr Vorgehen und Ihre Ergebnisse hinreichend. Sie sollten in der Lage sein, bei der Präsentation zu allen Teilaufgaben etwas sagen zu können.

2.1 Aufnahmebogen

Legen Sie eine Excel-Datei an und erstellen Sie für jede der vier Teilaufgaben (Ortsauswahl, Standardisierung, Versuchsplan, Steepest Ascent) ein separates Sheet. Erstellen Sie in jedem Sheet einen Aufnahmebogen, der mindestens über die folgenden Spalten verfügt: Wurf, Ort, Datum, Abwurfhöhe, Flugdauer. Die Aufnahmebögen für die Sheets Versuchsplan und Steepest Ascent enthalten zusätzlich die Spalten: Fluegellaenge, Koerperlaenge, Einschnitt, Papierstaerke. Achten Sie darauf, die Umlaute wie vorgeschrieben auszuschreiben. Achten Sie bei der Aufnahme der Flugdaten weiterhin darauf, dass Ihr Aufnahmebogen immer dem Prinzip der *Tidy Data* entspricht. Informationen zum Prinzip der *Tidy Data* finden Sie im Isis-Kurs von IDA.

2.2 Ortsauswahl

Fertigen Sie einen Papierhelikopter mit den Abmaßen des Referenzhelikopters an. Im ISIS-Kurs befindet sich eine Vorlage, die Sie nutzen können. Jeder Helikopter schließt mit einer Büroklammer ab. Führen Sie an vier Orten jeweils fünf Testflüge mit dem Helikopter durch und erfassen Sie die Flugzeiten im Aufnahmebogen Ortsauswahl.

Die Testflüge werden Ihnen bei den folgenden Punkten behilflich sein:

- einen geeigneten Ort für ihre Experimente zu finden (z. B. Haupt-, Mathe- oder EN-Gebäude, aber nicht die 3. Etage des Lichthofes.)
- potentielle Störgrößen zu identifizieren und daraufhin zu eliminieren

2.3 Standardisierung

- Berechnen und interpretieren Sie den c_p -Wert für Ihre Versuche aus Aufgabe 2.2. Die Toleranzbreite für die Zielgröße Flugdauer ist durch 0,5 s gegeben.
- Nutzen Sie ein Ursache-Wirkungs-Diagramm, um Größen zu identifizieren, die einen Einfluss auf die Zielgrößen Flugzeit und Stabilität (und somit die Streuung der Zielgröße) haben könnten.
- Entwickeln und dokumentieren Sie nun eine standardisierte Vorgehensweise, um eine möglichst geringe Streuung der Flugzeit zu erhalten. Die Dokumentation (in diesem Fall die Shiny-App) enthält dabei Aussagen / Anweisungen / Fotos bzgl. des Abwurfprozess / Abwurfeinrichtung / Zeitmessung etc., die es einem Außenstehenden ermöglicht, Ihren Versuchsaufbau in zuverlässiger Weise zu reproduzieren.
- Testen Sie Ihre standardisierte Vorgehensweise, indem Sie 20 Flüge unter standardisierten Bedingungen durchführen und wiederum den c_p -Wert berechnen und interpretieren. Erfassen Sie die Flugzeiten im Aufnahmebogen Standardisierung.

2.4 Vollfaktorieller Versuchsplan

- Führen Sie einen 2^4 -vollfaktoriellen Versuchsplan durch. Erstellen Sie hierzu einen randomisierten 2^4 -vollfaktoriellen Versuchsplan mit 2 Wiederholungen pro Faktorstufenkombination und 16 Versuchen im Zentrum (Centerpoints). Zur Erstellung eines Versuchsplanes mit 2 Replikationen und 16 Centerpoints können z.B. folgende Funktionen aus dem Package `qualityTools` genutzt werden:

```

1   if (!require(qualityTools)) {
2     install.packages("qualityTools")
3     require(qualityTools)
4   }
5
6   vp <- facDesign(k = 4, replicates = 2, centerCube =
7     16)
8   names(vp) <- c("Fluegellaenge", "Koerperlaenge", "
9     Einschnitt", "Papierart")
10  lows(vp) <- c(60, 50, 0, 80)
11  highs(vp) <- c(90, 100, 60, 120)
12  units(vp) <- c("mm", "mm", "mm", "g/mm^2")
13
14  summary(vp)

```

- Ermitteln Sie die Flugzeiten für die verschiedenen Faktorstufenkombinationen und tragen Sie sie in den Aufnahmebogen Versuchsplan ein. Die Reihenfolge der Versuche entnehmen Sie der Spalte `RunOrder`.

- (c) Untersuchen Sie die Haupt- und Wechselwirkungen der betrachteten Faktoren. Benutzen Sie dazu folgende Funktionen aus dem Package `qualityTools`, die auf Objekte der Klasse `facDesign` angewandt werden: `effectPlot()`, `interactionPlot()`, `paretoPlot()`. Orientieren Sie sich an diesen Plots bei der Erstellung der Plots für die Shiny-App mit `ggplot2`. Um die Funktionen nutzen zu können, müssen Sie das `facDesign`-Objekt um die gemessenen Flugzeiten erweitern:

```
1   flugzeiten <- c(4.3, 5.6, ...)
2   response(vp) <- flugzeiten
3
```

- (d) Erstellen und interpretieren Sie eine alle Haupt- und Wechselwirkungen enthaltende Regressionsgleichung. Nutzen Sie dazu die Funktion `lm`.
- (e) Vereinfachen Sie Ihr Regressionsmodell so, dass nur signifikante Haupt- und Wechselwirkungen enthalten sind. Falls Sie nicht mindestens zwei signifikante Haupteffekte erhalten, haben Sie die Störgrößen nicht unter Kontrolle, was vermutlich auf eine unzureichende Standardisierung zurückzuführen ist.
- (f) Vergleichen Sie den Intercept ihres Regressionsmodells mit den Flugzeiten des Referenzdesigns (0, 0, 0, 0).
- (g) Überprüfen Sie die zeitliche Abhängigkeit der Residuen des Referenzdesigns (0, 0, 0, 0).
- (h) Wozu dient die Randomisierung eines Versuchsplans? Welche Gefahren liegen in einer nicht-randomisierten Durchführung der Versuche?
- (i) Wie könnten Sie die hohe Anzahl der Versuche verringern, ohne weniger Faktoren betrachten zu müssen?

2.5 Steepest Ascent

- (a) Ermitteln Sie den Pfad des steilsten Anstieges (`qualityTools::steepAscent()`).
- (b) Fertigen Sie für ausgewählte Punkte entlang des steilsten Anstieges DAP-Copter an und ermitteln Sie jeweils die durchschnittliche Flugdauer.
- (c) Vergleichen Sie die tatsächliche Flugdauer mit der von Ihrem Modell vorhergesagten Flugdauer.
- (d) Um wie viel Prozent konnte die Flugzeit gegenüber dem Referenzmodell gesteigert werden?
- (e) Nehmen Sie kurz Stellung zu Ihren Ergebnissen (kritische Würdigung des Experimentes, Versuchsanordnung, Durchführung, mögliche Verbesserungen (Ablauf und Design), weitere Vorgehensweise etc., Probleme).

3 Entwicklung der Applikation

Durch die Bearbeitung der ersten Phase haben Sie die Optimierung eines spezifischen Copters durchgeführt. Um Ihre Ergebnisse zum einen zu präsentieren, aber auch Ihre dabei erworbenen Kenntnisse dem Unternehmen langfristig zur Verfügung zu stellen, sollen Sie nun eine Web-Applikation mit dem Package **shiny** entwickeln. Informationen zu **shiny** finden Sie hier. Falls Sie noch keine Erfahrung mit **shiny** haben, wird dringend empfohlen, den Umgang mit **shiny** an kleineren Applikationen zu üben und die verlinkte Webseite eingehend zu studieren.

3.1 Workflow

Mithilfe der bereits existierenden Import-Schnittstelle kann die .xlsx-Datei eingelesen werden und in `.values$data_storage` gespeichert werden. Die einzelnen Teilaufgaben werden durch vier Module repräsentiert, die als App in der App verstanden werden können. Jedem Modul wird `.values` übergeben. Zur Auswahl eines Datensatzes innerhalb dieser vier Module wird in jedem Modul das ebenfalls vordefinierte Modul `data_selector` eingebunden, das den ausgewählten Datensatz im Modul verfügbar macht. Dazu ist in jedem der vier Module bereits ein `reactive` mit Name `data` definiert, das den jeweils ausgewählten Datensatz enthält. Basierend auf dem ausgewählten Datensatz erzeugen Sie Ihre Visualisierungen. In den die vier Module repräsentierenden `tabItem` können dem Nutzer Eingabemöglichkeiten zur Verfügung gestellt werden. Plots und Tabellen werden im Viewer dargestellt (vgl. 3.5).

3.2 Dateien

Um mit der Programmierung der Shiny-App zu beginnen, laden Sie sich die Datei `app.zip` von ISIS herunter und entpacken Sie sie auf Ihrem Computer in einen Ordner Ihrer Wahl. Der Ordner enthält nun die folgenden Dateien und Ordner auf deren Funktion im Folgenden eingegangen wird:

- Dateien
 - App.R
 - Dashboard.Rproj
- Ordner
 - modules
 - * aufgaben
 - * predefined
 - www
 - * styles.css

Das R-Skript `App.R` ist der Einstiegspunkt in die App. Öffnen Sie `App.R` mit RStudio und klicken Sie 'Run App', um die App auszuführen. Wie jede Shiny-App besteht auch diese App aus zwei Teilen: dem `ui`-Element und der `server`-Funktion. Veränderungen an dieser Datei sind, solange sie keine grundlegenden Änderungen am Layout vornehmen oder weitere Module einbinden wollen, nicht notwendig.

Der Ordner `modules/aufgaben` enthält für jeden Aufgabenteil ein eigenes R-Skript, in dem Sie die Funktionalitäten für den jeweiligen Aufgabenteil programmieren können. Module werden mithilfe von `callModule` eingebunden und dienen dazu, eine komplexe App aus Gründen der Übersichtlichkeit und der Vermeidung von doppelter Variablenbenennung in mehrere Teile aufzuteilen. Weiterführende Informationen zu Shiny Modules finden Sie hier.

Weiterhin befinden sich in `modules/predefined` weitere R-Skripte, die z.B. den Import, die Speicherung und die Auswahl von Datensätzen definieren. Änderungen an diesen Skripten sollten nur mit Bedacht oder nach Rücksprache vorgenommen werden. Die in den einzelnen R-Skripten definierten Funktionen werden in `App.R` mithilfe der Funktion `source_directory` verfügbar gemacht.

Der Ordner `www` dient als Kommunikationsschnittstelle zum Client (Browser). Möchten Sie Sachen im Client rendern oder ausführen, die nicht direkt in einem R-Skript erzeugt werden, so legen Sie diese in jenem Ordner ab. Neben Bildern und Videos gehören dazu auch clientseitige Skripte wie z.B. CSS oder JavaScript, die das Layout oder auch die Funktionalität einer Shiny-App modifizieren. Dieser Ordner enthält bereits die CSS-Datei `styles.css`, die sie nach Belieben erweitern oder verändern können.

3.3 Anforderungen an die App

Die App muss in ihrer endgültigen Form mindestens über folgende Elemente verfügen. Alle Plots müssen mit `ggplot2` oder `plotly` erstellt werden und mit dem Viewer verknüpft werden. Die Plots aus `qualityTools` dienen nur zur Orientierung. Bauen Sie die Dokumentation Ihres Vorgehens entweder bei den einzelnen Aufgabenteilen ein oder erstellen Sie ein Modul, in dem die gesamte Dokumentation übersichtlich dargestellt wird.

- Import (bereits vorhanden)
- Projekt
 - Ortsauswahl
 - * Histogramm der Flugzeiten
 - Standardisierung
 - * Histogramm der Flugzeiten
 - * Ursache-Wirkungs-Diagramm (statisch, z.B. als .png oder .jpeg eingebunden)
 - * Foto von Abwurfprozess/Abwurfeinrichtung/Zeitmessung

- Versuchsplan
 - * Histogramm der Flugzeiten
 - * Effect Plot
 - * Interaction Plot (Hier reicht es, die Wechselwirkungen zwischen zwei Faktoren darzustellen, wobei diese Faktoren durch den Benutzer anpassbar sind)
 - * Pareto Plot
 - * Contour Plot / Surface Plot (Sowohl 2D als auch 3D; die Faktoren für x- und y-Achse müssen frei wählbar sein; Auswahlmöglichkeit, ob Wechselwirkung zwischen den Faktoren betrachtet wird)
- Steepest Ascent
 - * Steepest Ascent auf Contour Plot / Surface Plot (Sowohl 2D als auch 3D)
 - * Vergleich von für den Steepest Ascent vorhergesagten Flugzeiten mit den beobachteten Flugzeiten
- Gegebenenfalls Einstellungen, die z.B. das Layout der App anpassbar machen.

3.4 Erstellen eines neuen Moduls

Falls benötigt, können Sie im Ordner `modules` ein neues Modul anlegen. Orientieren Sie sich dazu an folgendem Minimalbeispiel:

```

1  module_name_ui <- function(id) {
2    ns <- NS(id)
3
4    tagList(
5      # content
6    )
7  }
8
9  module_name <- function(input, output, session, .values) {
10
11    ns <- session$ns
12  }
```

3.5 Weitere Hinweise

- Mit `.values$viewer$append_tab(tab = tabPanel())` wird dem Viewer ein `tabPanel` hinzugefügt. Der Viewer stellt eine Erweiterung einer `tabBox` dar, die z.B. gewährleistet, dass dasselbe `tabPanel` höchstens ein Mal in der `tabBox` enthalten ist und es ermöglicht einzelne `tabPanel` wieder zu schließen. Zur Orientierung wird in

`modules/aufgaben/ortsauswahl.R` beispielhaft ein Histogramm erstellt und über ein `actionButtonQW` im Viewer geöffnet.

- `actionButtonQW` erzeugt einen optisch ansprechenderen `actionButton`, der zusätzlich direkt mit einem `tooltip` ausgestattet werden kann.
- Um Namenskollisionen zu verhindern, besitzt jedes Modul einen eigenen Namespace, der mit `ns <- NS(id)` initialisiert wird. Jede `inputId` und `outputId` in der `ui`-Funktion muss mit `ns()` gewrapt werden. Bei der Verwendung von `renderUI()` in der `server`-Funktion müssen enthaltene `inputId` und `outputId` ebenfalls mit `ns()` gewrapt werden und in der `server`-Funktion folgende Zuweisung vorgenommen werden: `ns <- session$ns`
- Um input-Widgets reaktiv zu gestalten, gibt es die Möglichkeit diese mithilfe von `uiOutput()` und `renderUI()` in der `server`-Funktion zu definieren. Weitere Information finden Sie hier.
- Um interaktive Plots zu erstellen, empfiehlt sich die Benutzung von Funktionen aus dem Package `plotly`. Mit der Funktion `ggplotly()` können mit `ggplot2` erstellte Plots in `plotly`-Plots überführt werden. Beachten Sie, dass für die Darstellung von `plotly`-Plots `renderPlotly()` anstelle von `renderPlot()` bzw. `plotlyOutput()` anstelle von `plotOutput()` verwendet werden muss.

4 Präsentation der Case-Study

Laden Sie bis zum 19.07. folgende Dateien auf ISIS hoch:

- Die `.xlsx`-Datei, die alle Ihre Flugdaten, auf vier Sheets verteilt, enthält.
- Den kompletten Ordner als `.zip`-Datei, der Ihre Shiny-App inklusive aller zu ihrer korrekten Ausführung benötigten Dateien enthält.

In der Woche vom 22.07 bis zum 26.07 stellen Sie uns Ihre Projektarbeit etwa eine halbe Stunde lang vor. In der ersten Hälfte präsentieren Sie die Motivation, Durchführung und die Ergebnisse des Projektes mithilfe der von Ihnen erstellten App, wobei Sie die `.xlsx`-Datei verwenden, die Sie zur Aufnahme Ihrer Flugdaten verwendet haben. In der zweiten Hälfte werden Sie eine `.xlsx`-Datei mit Flugdaten in identischer Struktur von uns bekommen und die Daten mithilfe Ihrer App analysieren. Gestalten Sie den Inhalt der App so umfangreich wie möglich.

Die Vorstellung der App erfolgt über einen Beamer. Um sicherzustellen, dass alles wie gewünscht funktioniert, empfehlen wir Ihnen, dass sie sich in den Tagen vor der Präsentation mit der Technik im Raum PTZ 407 vertraut machen.