ЦКП. Свойства факторных экспериментов. Критерии оптимальности планов

Центральный композиционный план

Если по каким-либо причинам линейная модель не соответствует задачам, то планы второго порядка позволяют сформировать функцию отклика в виде полного квадратичного полинома, который содержит большее число членов, чем неполный квадратичный полином, сформированный по планам первого порядка. Такие планы требуют большее число выполняемых опытов.

$$y = b_0 + b_1x_1 + b_2x_2 + b_{12}x_1x_2 + \ldots + b_{11}x_1^2 + \ldots$$

Для получения квадратичной зависимости каждый фактор должен фиксироваться как минимум на трех уровнях. Для планов второго порядка область планирования может:

- быть естественной - включать область планирования планов первого порядка и дополнительные точки (**такие планы называются композиционными**). Дополнительные точки могут выходить за область плана первого порядка — единичного гиперкуба. В этом случае опыты в них реализуются при установлении факторов за пределами варьирования. Это надо учитывать при определении области совместимости факторов;
- не выходить за пределы единичного гиперкуба;
- не выходить за пределы единичного гипершара;

В **ЦКП** переменные варьируются на пяти уровнях:

- 2 уровня характерные для ПФЭ
- 2 дополнительных уровня за счет добавления "звездных точек"
- 1 дополнительная центральная точка

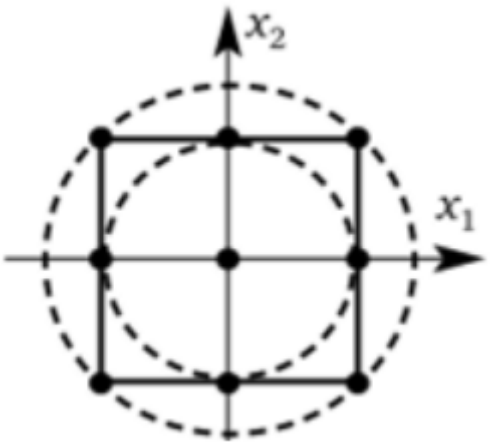
К примеру для фактора ПФЭ 2^3:

Матрица ЦКП таким образом будет состоять из композиции матрицы ПФЭ 2^3 и дополнитлеьной матрицы звездных и нулевых точек (где α - какое-то число).

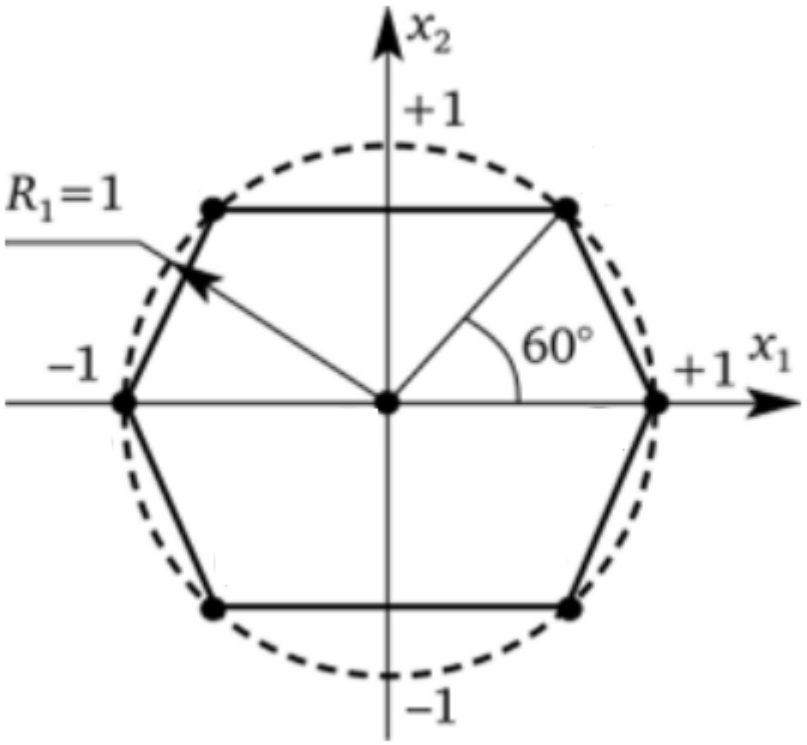
−	−	−
+	−	−
−	+	−
+	+	−
−	−	+
+	−	+
−	+	+
+	+	+
$-\alpha_1$	0	0
α_1	0	0
0	$-\alpha_2$	0
0	α_2	0
0	0	$-\alpha_3$
0	0	α_3
0	0	0
0	0	0

Центральные композиционные планы также строятся с условием соблюдения **ортогональности** и **ротабельности**.

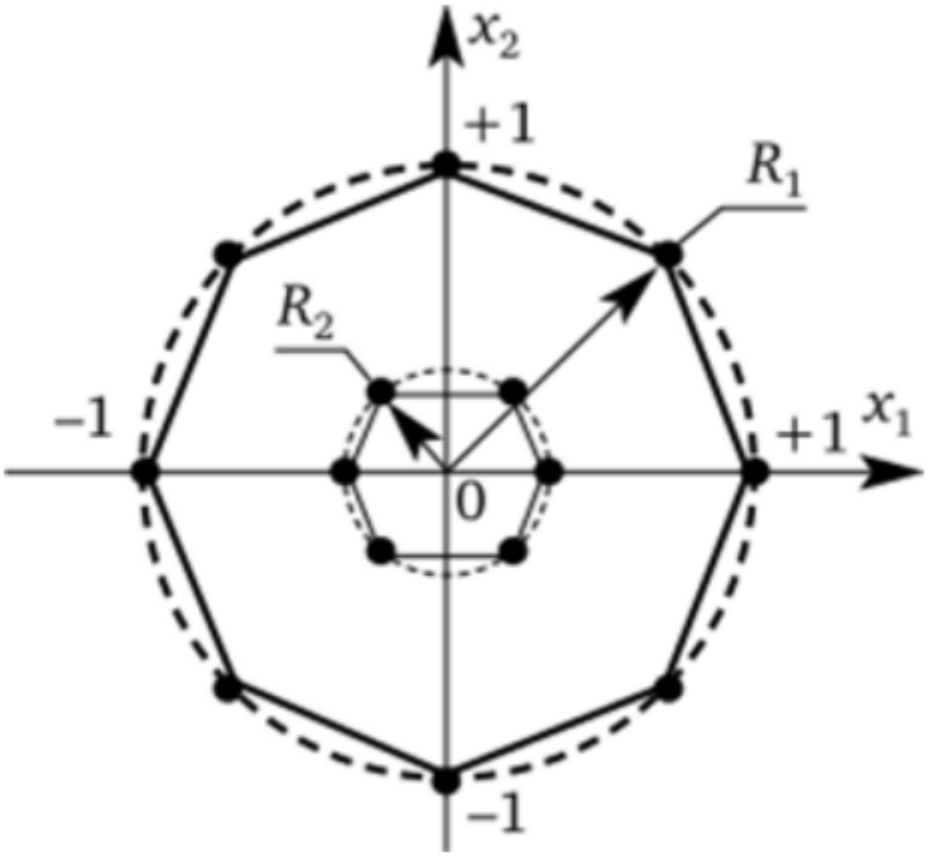
Пример не ротабльного плана:



Пример ротабельного плана:



Пример ротабельного плана с двумя концентрическими окружностями:



ПФЭ

$$N = n^k$$

где n - количество уровней; k - число факторов.

$$N_{\text{ЦКП}} = N + 2k + N_0$$

где N_0 - количество точек в центре плана.

Реплики

$$N = n^{k-p}$$

где p - порядок реплики (p = 1 для полуреплики).

для ОЦКП

$$\alpha = 2^{k/4}$$

при k < 5

$$\alpha = \sqrt{\sqrt{2^{k-p-2}}(\sqrt{N} - \sqrt{2^{k*p}})}$$

или

$$\alpha = N^{1/4}$$

для РЦКП

$$\alpha = 2^{k/4}$$

при k < 5

$$\alpha = 2^{(k-1)/4}$$

при k >= 5

N_0 может быть 1, для ОЦКП и "сбалансированным" (>1) для РЦКП (по таблице).

Для соблюдений ортогональности и ротабельности количество может быть определено как:

$$N_0 = 4\sqrt{N} + 4 - 2k$$

Ref:

- Khuri and Cornell, 1987
- StatSoft: Design of Experiments: Science, Industrial DOE

Число факторов, n	Число опытов в ядре плана, $N_я$	Величина «звездного плеча», α	Число опытов в»звездных точках», $2n$	Число опытов в центре плана, N_0	Общее число опытов, N
2	2^2 (ПФЭ)	1,414	4	5	13
3	2^3 (ПФЭ)	1,682	6	6	20
4	2^4 (ПФЭ)	2,000	8	7	31
5	2^5 (ПФЭ)	2,378	10	10	52
5	2^{5-1}	2,000	10	6	32
6	(ДФЭ)	2,828	12	15	91
6	2^6 (ПФЭ)	2,378	12	9	53
7	2^{6-1}	3,333	14	21	163
7	(ДФЭ)	2,828	14	14	92
	2^7 (ПФЭ)				
	2^{7-1}				
	(ДФЭ)				

Параметр плана.	Количество факторов								
	2	3	4	5	5	6	6	7	7
Ядро плана	2^2	2^3	2^4	2^5	2^{5-1}	2^6	2^{6-1}	2^7	2^{7-1}
α	1.414	1.682	2.0	2.378	2.0	2.83	2.38	3.36	2.83
n_0	5	6	7	10	6	15	9	21	14

In [1]:
```
k = 2
N = 4
4sqrt(N) + 4 - 2k
```

Out[1]: 8.0

Повторение: некоторые свойства факторных экспериментов (совойства плана)

1. **Ортогональность**: скалярные произведения столбцов равны 0. Иными словами, сумма построчных парных произведений чисел первого столбца (для x_1) на соответствующие числа второго столбца (x_2) равна 0 - и так далее. Это говорит об отсутствии взаимодействия между столбцами, поэтому коэффициенты уравнения регрессии определяются независимо друг от друга, что очень важно для интерпретации коэффициентов уравнения. Такие планы называются D-оптимальными.

2. **Симметричность**: число плюсов равно числу минусов по каждому столбцу. Все коэффициенты уравнения определяются с минимальными и равными ошибками.

3. **Рототабельность (свойство нормировки)**: сумма квадратов элементов столбцов равна числу опытов. Они позволяют рассчитывать поверхность отклика с одинаковыми ошибками для всех точек факторного пространства, равноудалённых от центра эксперимента.

Определения:

Информационной матрицей плана эксперимента называется матрица:

$$M = (X^T \cdot X)$$

Где X - дизайн-матрица (развернутая матрица плана).

Ковариационная матрица оценок параметров:

$$\Sigma = \sigma^2 M^{-1} = \sigma^2 (X^T \cdot X)^{-1}$$

А **дисперсионная матрица** плана:

$$\mathrm{C} = M^{-1} = (X^T \cdot X)^{-1}$$

Свойства информационной матрицы:

- Матрица M — симметричная и положительно определенная матрица.
- Если ранг матрицы M есть $rank(M) = k$, то по имеющимся данным можно оценить ровно k линейно независимых линейных комбинаций параметров модели.

Наилучший выбор плана D при проведении активного эксперимента можно осуществить различными способами в зависимости от целей эксперимента. Иными словами, критерии оптимальности плана могут быть различными.

Различают три группы критериев оптимальности плана:

1. критерии, связанные со свойствами оценок коэффициентов модели;
2. критерии, связанные с предсказательными свойствами модели, т.е. с дисперсией отклика;
3. критерии, формулируемые без использования матрицы плана и дисперсионной матрицы.

Группа 1

- **D-оптимальность**

Минимум обобщенной дисперсии.

Минимум определителя дисперсионной матрицы, пусть D - некоторый план, а Ω - некоторое множество возможных вариантов варьирования, тогда D-оптимальность определяется выражением:

$$opt = \min_{D \in \Omega} |C(D)| = \min_{D \in \Omega} |(X^T \cdot X)^{-1}|$$

где $|C(D)| = det(C(D))$ - определеитель матрицы C плана D.

Или:

$$opt = \max_{D \in \Omega} |M(D)| = \max_{D \in \Omega} |(X^T \cdot X)|$$

In [11]:
```
using LinearAlgebra, Statistics

D1 = [1 -1; 1 1; -1 1; -1 -1]
D2 = [1 -1; 0 1; -1 1; -1  0]
M1 = D1' * D1
M2 = D2' * D2
println("Det M1 = " , det(M1), ", Det M2 = ", det(M2))
```

```
Det M1 = 16.0, Det M2 = 5.0
```

- **A-оптимальность**

Минимальная средняя дисперсия оценок коэффициентов.

Минимум следа дисперсионной матрицы:

$$opt = \min_{D \in \Omega} tr(C)$$

или

$$opt = \min_{D \in \Omega} tr(\mathrm{M}^{-1})$$

In [3]:
```
println("Trace M1⁻¹ = " , tr(inv(M1)), ", Trace M2⁻¹ = ", tr(inv(M2)))
```

```
Trace M1⁻¹ = 0.5, Trace M2⁻¹ = 1.1999999999999997
```

- **E-оптимальность**

Отдельные оценки параметров не обладают слишком большими дисперсиями.

Максимум минимального значния собственных чисел информационной матрицы или минимум максимального значения собственных чисел дисперсионной матрицы:

$$opt = \min \max \lambda(C(D))$$

$$opt = \max \min \lambda(M(D))$$

In [4]:
```
println("max min λ M1   = " , minimum(eigvals(M1)),      ",  max min λ M2   = ",
println("min max λ M1⁻¹ = " , maximum(eigvals(inv(M1))), ", min max λ M2⁻¹ = ",
```

```
max min λ M1   = 4.0,  max min λ M2   = 0.9999999999999998
min max λ M1⁻¹ = 0.25, min max λ M2⁻¹ = 0.9999999999999998
```

- **Ортогональность**

Оценки параметров независимы $cov\{\beta_j\ \beta_u\} = 0$

Диагональность дисперсионной матрицы.

In [5]:
```
M1
```

Out[5]:
```
2×2 Matrix{Int64}:
 4  0
 0  4
```

In [6]:
```
M2
```

Out[6]:
```
2×2 Matrix{Int64}:
  3  -2
 -2   3
```

Для линейных моделей 1-го порядка ортогональный план (при котором $X^T X = NE$, где E - единичная матрица) является оптимальным по всем другим критериям. В связи с этим, планирование эксперимента сводится к изучению ортогональных планов.

Матрица X состоящая из $+1$ и -1 и удовлетворяющая этому условию называется матрицей Адамара.

Группа 2

- **G-оптимальность**

Минимум максимального значения дисперсии оценок отклика.

$$opt = \min\max diag(C) = \min\max diag(M^{-1})$$

- **Q-оптимальность**

Минимум средней дисперсии отклика.

- **Ротабельность**

Постоянство дисперсии предсказания на равных расстояниях от центра эксперимента.

Группа 3

- **Насыщенность**

Степень близости общего числа опытов к числу неизвестных параметров модели.

- **Композиционность**

Свойство плана, означающее возможность разделить эксперимент на части и реализовать каждую часть последовательно.

Существуют и другие виды оптимальности планов (S-оптимальность, T-оптимальность и т.д.).

Построение оптимальных планов

Алгоритмы построения оптимальных планов обычно заключаются в поиске

определенного числа точек из числа "точек-кандидатов", которые максимизируют качество плана по определенному критерию.

Такой "поиск" наилучшего плана не является точным метод, а алгоритмическая процедура, использующая некоторые стратегии поиска для нахождения наилучшего плана (согласно некоторому критерию оптимальности).

Далее приведены некоторые методы (Cook и Nachtsheim, 1980).

Последовательный метод или метод Дейкстры (Dykstra, 1971): начиная с пустого плана, алгоритм ведет поиск по списку "точек-кандидатов" и на каждом шаге отбирает одну, которая максимизирует выбранный критерий. Это самый быстрый из обсуждаемых, но не гарантирует выбор оптимального плана.

Метод простого обмена или метод Винна-Митчелла (Mitchell и Miller (1970) и Wynn (1972)): метод начинается с начального плана требуемого объема (для формирования обычно используется метод Дейкстры). В каждой итерации одна точка удаляется из плана, а на ее место включается точка из списка кандидатов. Выбор точек для удаления и включения последовательный, то есть на каждом шаге точка, добавляющая меньше всего относительно выбранного критерия оптимальности выбрасывается из плана, затем алгоритм отбирает точку из списка кандидатов для максимального увеличения соответствующего критерия.

Алгоритм DETMAX - обмен с отклонениями (Mitchell, 1974b): вначале строится исходный план (с помощью метода простого обмена). Поиск начинается с применения алгоритма простого обмена, однако, если соответствующий критерий оптимальности не улучшается, алгоритм предпринимает отклонения - т.е. добавляет или выбрасывает более одной точки за один раз, так что во время поиска число точек в плане может изменяться между $N_D - N_{\text{откл}}$ и $N_D + N_{\text{откл}}$, где N_D – требуемый объем плана, а $N_{\text{откл}}$ - максимально допустимое отклонение, определяемое пользователем.

Модифицированный алгоритм Федорова (одновременного переключения) (Cook и Nachtsheim, 1980): начинается с исходного плана требуемого объема. На каждой итерации алгоритм обменивается каждой точкой плана с отобранной из списка "точек-кандидатов". В отличие от алгоритма простого обмена, в данном алгоритме обмен не последовательный, а одновременный. Так, на каждой итерации каждая точка плана сравнивается с каждой точкой из списка кандидатов, и обмен происходит парой, оптимизирующей план.

Алгоритм Федорова (одновременного переключения): отличие этого алгоитма от модифицированного алгоритма Федорова, заключается в том, что на каждой итерации осуществляется только единственный обмен, то есть на каждой итерации оцениваются все возможные пары точек плана и списка кандидатов. Алгоритм обменивается парой, оптимизирующей план относительно выбранного критерия.

К сожалению не существует точного решения проблемы оптимального плана.т.к. детерминант матрицы $X^T X$ (и след ее обратной $tr((X^T X)^{-1})$ являются сложными

функциями списка точек-кандидатов, то может существовать несколько локальных минимумов этой функции.

Следовательно, важно изучит несколько начальных планов и несколько алгоритмов. Если после повторения оптимизации несколько раз со случайного старта получится тот же самый или близкий оптимальный план, тогда вы можете быть в достаточной мере уверены, что вы не "застряли" в локальном минимуме или максимуме.

In [7]:
```
using ExperimentalDesign, StatsModels, GLM, DataFrames, Distributions, Random

design_distribution = DesignDistribution((f1 = DiscreteUniform(0, 5),
        f2 = CategoricalFactor(["cf", "cg", "ca"])))
```

Out[7]:
```
DesignDistribution
Formula: 0 ~ f1 + f2
Factor Distributions:
f1: DiscreteUniform(a=0, b=5)
f2: CategoricalFactor(
values: ["cf", "cg", "ca"]
distribution: DiscreteUniform(a=1, b=3)
)
```

In [8]:
```
design = rand(design_distribution, 300)
```

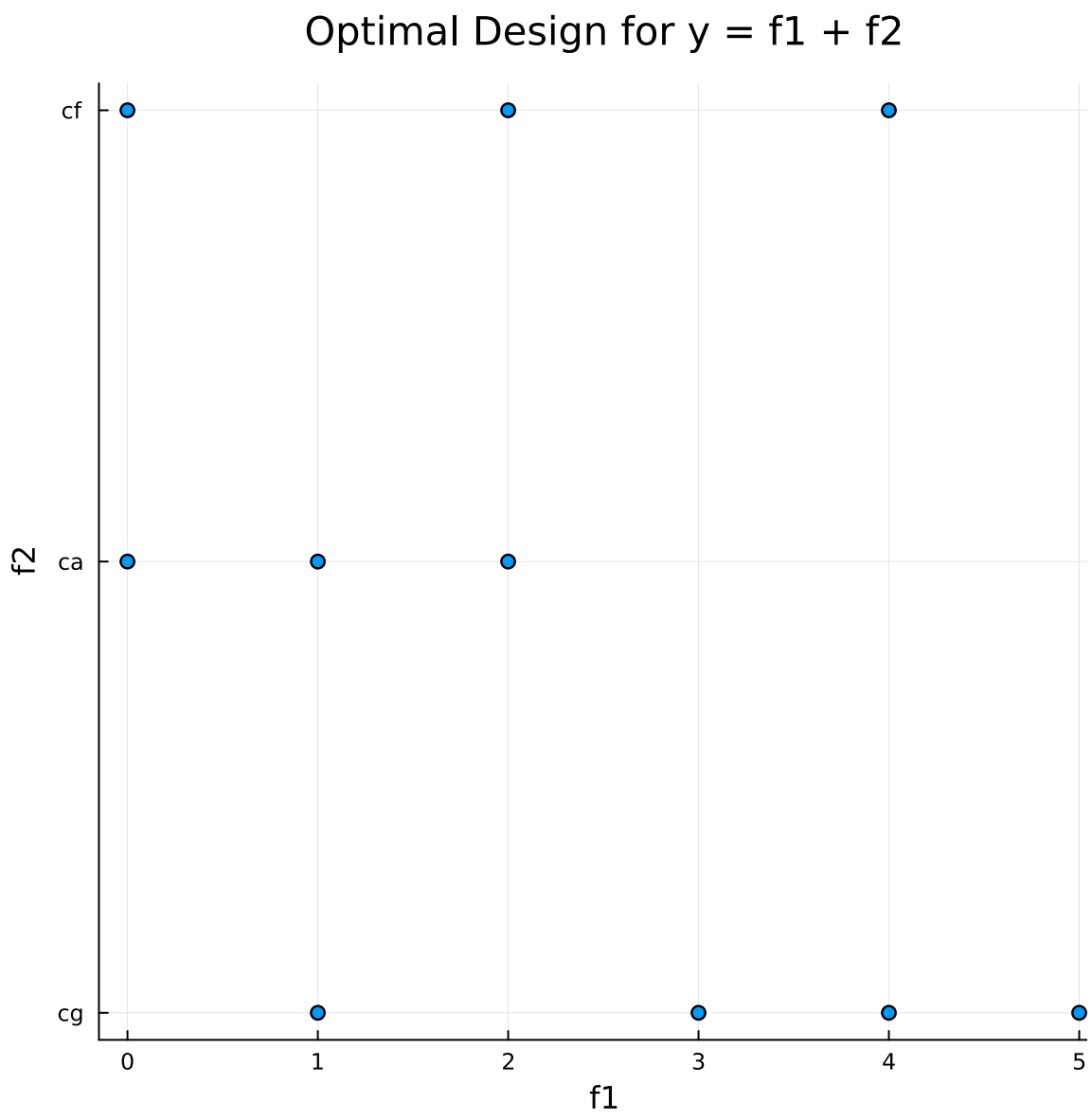
Out[8]:
```
ExperimentalDesign.RandomDesign
Dimension: (300, 2)
Factors: (f1 = DiscreteUniform(a=0, b=5), f2 = CategoricalFactor(
values: ["cf", "cg", "ca"]
distribution: DiscreteUniform(a=1, b=3)
)
)
Formula: 0 ~ f1 + f2
Design Matrix:
300×2 DataFrame
 Row │ f1    f2
     │ Any   Any
─────┼──────────
   1 │ 1     ca
   2 │ 5     cf
   3 │ 0     ca
   4 │ 4     ca
   5 │ 1     ca
   6 │ 5     cg
   7 │ 0     cg
   8 │ 5     ca
   9 │ 5     ca
  10 │ 3     cg
  11 │ 2     cf
   ⋮ │  ⋮    ⋮
 291 │ 0     ca
 292 │ 3     cf
 293 │ 0     cf
 294 │ 5     cf
 295 │ 2     ca
 296 │ 5     ca
 297 │ 4     cf
 298 │ 1     cg
 299 │ 5     cf
 300 │ 3     ca
279 rows omitted
```

In [16]:
```
using StatsPlots
f = @formula 0 ~ f1 + f1 ^ 2 + f2
optimal_design = OptimalDesign(design, f, 10)
```

Out[16]:
```
OptimalDesign
Dimension: (10, 2)
Factors: (f1 = DiscreteUniform(a=0, b=5), f2 = CategoricalFactor(
values: ["cf", "cg", "ca"]
distribution: DiscreteUniform(a=1, b=3)
)
)
Formula: 0 ~ f1 + :(f1 ^ 2) + f2
Selected Candidate Rows: [85, 105, 102, 136, 134, 36, 21, 10, 1, 27]
Optimality Criteria: Dict(:D => 0.029985918141278503)
Design Matrix:
10×2 DataFrame
 Row │ f1    f2
     │ Any   Any
─────┼──────────
   1 │ 5     cg
   2 │ 1     cg
   3 │ 2     ca
   4 │ 0     cf
   5 │ 4     cf
   6 │ 0     ca
   7 │ 2     cf
   8 │ 3     cg
   9 │ 1     ca
  10 │ 4     cg
```

In [17]:
```
@df optimal_design.matrix scatter(:f1,
    :f2,
    size = (600, 600),
    xlabel = "f1",
    ylabel = "f2",
    legend = false,
    title = "Optimal Design for y = f1 + f2")
```

Out[17]:



Материал для самоподготовки

Когда желательно/необходимо применение стандартизации и нормализации?

- Что такое стандартизация?

Стандартизация - это приведение даннных к единому масшабу таким образом, что получаем следующую статистику распределения данных:

Среднее: $\mu = 0$

Стандартное отклонение: $\sigma = 1$

Используя следующий вид:

$z = \frac{x-\mu}{\sigma}$

Стандартизация используется для значений данных, которые имеют нормальное распределение. Кроме того, применяя стандартизацию, мы стремимся сделать среднее значение набора данных равным 0, а стандартное отклонение эквивалентным 1.

Таким образом, набор данных становится более очевидным и удобным для анализа, а оценки получаются несмещенными.

Нормализация — это преобразование данных к неким безразмерным единицам. Обычно нормализация выполняется в рамках заданного диапазона, например, [0..1] или [-1..1].

К примеру:

$$\hat{x_i} = \frac{x_i - (max_x + min_x)/2}{(max_x - min_x)/2}$$

- Для чего применяется стандартизация

Для того чтобы наша модель работала хорошо, очень важно, чтобы данные имели одинаковый масштаб с точки зрения функции, чтобы избежать смещения в результате. Так, например, имея множество числовых независимых переменных, нам необходимо представить их в одном масштабе, чтобы сделать правильные выводы. В противном случае, разброс может получится максимальным, что усложнит выводы.

- Какие виды стандартизации бывают

Есть несколько способов стандартизации. Можно использовать функцию StandardScaler из библиотеки sklearn.preprocessing. Можно делать это по формуле, приводя среднее к нулю, а стандартное отклонение к единице.

Литература по разделу

- Сидняев Н. И., Теория планирования эксперимента и анализ статистических данных, ISBN 978-5-534-05070-7
- A. C. Atkinson and A. N. Donev, Optimum Experimental Designs, Oxford University Press, 1992.
- S. D. Silvey, Optimal Design, Chapman and Hall, 1980. (Just 86 pages but unfortunately out of print.)
- F. Pukelsheim, Optimal Design ofExperiments , Chapman and Hall, 1995.

Ссылки

- **Julia**
  - Ссылка: https://julialang.org/
- **Документация к основным пакетам Julia**
  - *Математика и анализ*

        - Roots.jl (нахождение корней) https://juliamath.github.io/Roots.jl/stable/
        - Optim.jl (Поиск минимума/максимума) https://julianlsolvers.github.io/Optim.jl/stable/#user/minimization/
        - ForwardDiff.jl (Дифференцирование) https://juliadiff.org/ForwardDiff.jl/stable/
        - QuadGK.jl (Численное интегрирование) https://juliamath.github.io/QuadGK.jl/stable/
        - DifferentialEquations.jl (Численное решение дифференциальных уравнений) https://diffeq.sciml.ai/stable/
        - ExperimentalDesign.jl (Оптимизация дизайна) https://github.com/phrb/ExperimentalDesign.jl
    - *Статистика*
        - Distributions.jl (Основной пакет для работы с распределениями) https://juliastats.org/Distributions.jl/stable/
        - Statistics - Базовый пакет с основными статистическими функциями (не требует установки, но надо подключать `using Statistics`)
        - HypothesisTests.jl (основные статистические критерии) https://juliastats.org/HypothesisTests.jl/stable/
        - GLM.jl (Общая/обобщенная линейная модель) https://juliastats.org/GLM.jl/stable/
    - *Другое*
        - Plots.jl (Графики) https://docs.juliaplots.org
- **Литература**
    - Математика. Базовый курс / Б.Ш. Гулиян, Р.Я. Хамидуллин. Москва : Синергия, 2013. - 712 с. - ISBN 978-5-4257-0109-1.
    - Фадеева Л. Н., Лебедев А. В., Теория вероятностей и математическая статистика: учебное пособие. - 2-е изд., перераб. и доп. - М.: Эксмо, 2010. - 496 с.
    - The Matrix Cookbook
    - Линейная алгебра и ее применения - Стренг Г.
    - Матричный анализ - Хорн Р., Джонсон Ч.
- **Дополнительная литература:**
    - Математическая статистика в медицине, В. А. Медик, М. С. Токмачев,

978-5-279-03195-5

- *Медико-биологическая статистика. Гланц. Пер. с англ. — М., Практика, 1998. — 459 с.*
- *Байесовская статистика: Star Wars, LEGO, резиновые уточки и многое другое*
- *Занимательная статистика. Манга. Син Такахаси, 2009, 224с, ISBN: 978-5-97060-179-2*

In [ ]: