DSA 8020 Project II

Meredith Sliger

Problem 1 Regression Modeling: Generalized Linear Model vs Linear Regression Model

Perform a detailed regression analysis using the `gala` data set. Use `Species` as the response variable, excluding `Endemics` from this analysis, to address the following questions:

1.1 Generalized Linear Model (GLM)

Conduct a regression analysis using one of the generalized linear models we covered in Week 7 (i.e., logistic regression or Poisson regression) and perform model selection.

1.1.1 Load and Prepare Data

```
# Load required packages
library(faraway)
library(dplyr)

# Load and inspect data
data(gala)
gala_mod <- gala %>% select(-Endemics)
head(gala_mod)
```

```
##              Species  Area Elevation Nearest Scruz Adjacent
## Baltra            58 25.09       346     0.6   0.6     1.84
## Bartolome         31  1.24       109     0.6  26.3   572.33
## Caldwell           3  0.21       114     2.8  58.7     0.78
## Champion          25  0.10        46     1.9  47.4     0.18
## Coamano            2  0.05        77     1.9   1.9   903.82
## Daphne.Major      18  0.34       119     8.0   8.0     1.84
```

1.1.2 Poisson Regression

```
glm_poisson <- glm(Species ~ ., family = poisson(link = "log"), data = gala_mod)
summary(glm_poisson)
```

```
##
## Call:
## glm(formula = Species ~ ., family = poisson(link = "log"), data = gala_mod)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
## Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
## Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
## Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
## Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
## Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  716.85  on 24  degrees of freedom
## AIC: 889.68
##
## Number of Fisher Scoring iterations: 5
```

1.1.3 Model Selection (Stepwise AIC)

```
glm_poisson_step <- step(glm_poisson, direction = "both")
```

```
## Start:  AIC=889.68
## Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##
##             Df Deviance     AIC
## <none>           716.85  889.68
## - Nearest    1   739.41  910.24
## - Scruz      1   813.62  984.45
## - Area       1  1204.35 1375.18
## - Adjacent   1  1341.45 1512.29
## - Elevation  1  2389.57 2560.40
```

```
summary(glm_poisson_step)
```

```
##
## Call:
## glm(formula = Species ~ Area + Elevation + Nearest + Scruz +
##     Adjacent, family = poisson(link = "log"), data = gala_mod)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.155e+00  5.175e-02  60.963  < 2e-16 ***
## Area        -5.799e-04  2.627e-05 -22.074  < 2e-16 ***
## Elevation    3.541e-03  8.741e-05  40.507  < 2e-16 ***
## Nearest      8.826e-03  1.821e-03   4.846 1.26e-06 ***
## Scruz       -5.709e-03  6.256e-04  -9.126  < 2e-16 ***
## Adjacent    -6.630e-04  2.933e-05 -22.608  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 3510.73  on 29  degrees of freedom
## Residual deviance:  716.85  on 24  degrees of freedom
## AIC: 889.68
##
## Number of Fisher Scoring iterations: 5
```

Interpretation

Since the response variable `Species` represents count data (number of species), a Poisson regression is appropriate. This model assumes that the response variable follows a Poisson distribution and uses a log link to relate the expected count to a linear combination of predictors. The model ensures all predicted values are non-negative and can properly account for the typical right-skewed distribution of counts.

The Poisson model was fitted using all predictors except `Endemics`, and stepwise AIC was used for model selection. The final model retained all five predictors (`Area`, `Elevation`, `Nearest`, `Scruz`, `Adjacent`) and resulted in a substantial drop in residual deviance compared to the null model, indicating a much better fit. The residual deviance was 716.85 with 24 degrees of freedom, compared to a null deviance of 3510.73.

1.2 Linear Regression Model

Perform a linear regression analysis and conduct model selection again to determine the most appropriate linear regression model. Compare it with the GLM model selected in the previous question and comment on your findings.

1.2.1 Linear Model Fit

```
lm_model <- lm(Species ~ ., data = gala_mod)
summary(lm_model)
```

```
##
## Call:
## lm(formula = Species ~ ., data = gala_mod)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -111.679  -34.898   -7.862   33.460  182.584
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.068221  19.154198   0.369 0.715351
## Area        -0.023938   0.022422  -1.068 0.296318
## Elevation    0.319465   0.053663   5.953 3.82e-06 ***
## Nearest      0.009144   1.054136   0.009 0.993151
## Scruz       -0.240524   0.215402  -1.117 0.275208
## Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.98 on 24 degrees of freedom
## Multiple R-squared:  0.7658, Adjusted R-squared:  0.7171
## F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

1.2.2 Model Selection (Stepwise AIC)

```r
lm_model_step <- step(lm_model, direction = "both")
```

```
## Start:  AIC=251.93
## Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## - Nearest    1         0  89232 249.93
## - Area       1      4238  93469 251.33
## - Scruz      1      4636  93867 251.45
## <none>                    89231 251.93
## - Adjacent   1     66406 155638 266.62
## - Elevation  1    131767 220998 277.14
##
## Step:  AIC=249.93
## Species ~ Area + Elevation + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## - Area       1      4436  93667 249.39
## <none>                    89232 249.93
## - Scruz      1      7544  96776 250.37
## + Nearest    1         0  89231 251.93
## - Adjacent   1     72312 161544 265.74
## - Elevation  1    139445 228677 276.17
##
## Step:  AIC=249.39
## Species ~ Elevation + Scruz + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## - Scruz      1      6336 100003 249.35
## <none>                    93667 249.39
## + Area       1      4436  89232 249.93
## + Nearest    1       198  93469 251.33
## - Adjacent   1     69860 163527 264.11
## - Elevation  1    275784 369451 288.56
##
## Step:  AIC=249.35
## Species ~ Elevation + Adjacent
##
##             Df Sum of Sq    RSS    AIC
## <none>                   100003 249.35
## + Scruz      1      6336  93667 249.39
## + Area       1      3227  96776 250.37
## + Nearest    1      1550  98453 250.88
## - Adjacent   1     73251 173254 263.84
## - Elevation  1    280817 380820 287.47
```

```r
summary(lm_model_step)
```

```
##
## Call:
## lm(formula = Species ~ Elevation + Adjacent, data = gala_mod)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -103.41  -34.33  -11.43   22.57  203.65
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.43287   15.02469   0.095 0.924727
## Elevation    0.27657    0.03176   8.707 2.53e-09 ***
## Adjacent    -0.06889    0.01549  -4.447 0.000134 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.86 on 27 degrees of freedom
## Multiple R-squared:  0.7376, Adjusted R-squared:  0.7181
## F-statistic: 37.94 on 2 and 27 DF,  p-value: 1.434e-08
```

Interpretation

A traditional linear regression model was also fitted using the same predictors. The initial model included all variables, and stepwise AIC selection led to a reduced model containing only `Elevation` and `Adjacent`. This model had a respectable adjusted R-squared of 0.7181 and a residual standard error of approximately 60.86.

However, linear regression assumes constant variance and normally distributed residuals. It also allows for the possibility of negative predictions, which is not meaningful for a count response. These limitations make it less suitable for modeling species richness compared to a Poisson regression.

1.3 Comparison Between GLM and Linear Model

While both models identify similar key predictors, the Poisson regression model is more appropriate for count data. Its log link function allows for multiplicative interpretation of predictor effects and restricts predictions to the non-negative range. The linear model, although interpretable in an additive sense, does not inherently respect the non-negativity and variance properties of count data.

Therefore, although the linear model fits reasonably well, the Poisson model better respects the data's structure and yields more reliable and interpretable results in this context. The final recommendation is to use the Poisson regression model for analyzing the number of species on the islands.

Problem 2: Paper Helicopter Experiment

2

A researcher would like to investigate how long a paper 'helicopter' can stay in the air when dropped from a height of 2 meters. Some potential treatment factors are: 1) Paper type (light, medium, and heavy); 2) Rotor length (7.5cm or 8.5cm); 3) Leg length (7.5cm or 12cm); 4) Leg width (3.2cmor 5cm). Answer the following questions:

2.1 Suppose the researcher would like to compare the responses between different paper types while keeping other factors fixed. What are the experimental units and measurement units in this study? Which design can she apply?

Experimental Unit: The experimental unit is the individual paper helicopter — not the flight time or seconds. This is the unit to which the treatment (paper type) is applied independently. The key distinction is that you randomize treatments to helicopters, not to seconds or time trials. This is consistent with the formal definition: the experimental unit is the smallest unit that can independently receive a different treatment.

Measurement Unit: The measurement unit is the observed flight time for each helicopter. This is what you measure, but you do not apply treatments to time — this is what the footnote in the project instructions warns against.

Design: Because the researcher is comparing paper types while keeping other factors constant, and each helicopter is independently assigned to a treatment, the appropriate design is a **Completely Randomized Design (CRD)**. This allows for unbiased comparisons among paper types and is straightforward to implement when blocking is unnecessary.

2.2 Suppose the researcher would like to conduct an experiment in (a) using 15 paper helicopters. Explain how she can perform the randomization and replication procedures here.

To compare the effects of three different **paper types** while keeping all other factors constant, the researcher has 15 paper helicopters to work with. A good design would involve **equal replication** across treatment levels to ensure balance and improve the precision of treatment effect estimates.

1. **Replication:**
   Since there are three paper types (e.g., light, medium, heavy), assign **5 helicopters to each paper type**. This gives equal replication:

   $$3 \text{ paper types} \times 5 \text{ replicates} = 15 \text{ helicopters}$$

2. **Randomization:**
   Randomization should be used to assign each helicopter to a paper type in a way that avoids bias:

   - Label the helicopters 1 through 15.
   - Randomly assign five helicopters to each paper type using a random number generator, software (e.g., `sample()` in R), or a randomization table.
   - Ensure the order of testing (e.g., which helicopter is dropped first) is also randomized if possible, to eliminate potential order effects.

This procedure satisfies two key principles of experimental design: - **Replication**, to estimate variability and increase statistical power. - **Randomization**, to protect against bias and confounding from unmeasured factors.

By applying treatments independently to each helicopter and using proper randomization, the researcher ensures the validity of the statistical inference from the experiment.

2.3 Think of a situation in which the researcher would need to consider using a randomized complete block design (RCBD).

A randomized complete block design (RCBD) should be used when there is a **known source of variability** that is not of primary interest but could affect the response. In this context, that means identifying a factor that might influence the helicopter's flight time but is not being intentionally studied.

Example Scenario:
Suppose the experiment is conducted over multiple sessions or by multiple testers. Environmental factors like **air currents, room temperature**, or **differences in how testers drop the helicopters** could introduce variability into the results. These are not treatment factors, but they could influence flight time.

In this case, the researcher can define **blocks** based on these sources of variation. For example: - Each tester is a block. - Each time slot (e.g., morning vs. afternoon) is a block.

Within each block, the researcher would randomly assign **all treatment levels** (e.g., all three paper types). This ensures that comparisons between treatments are made under similar conditions, which helps to: - Reduce unexplained variation. - Increase the sensitivity of the analysis to detect actual treatment effects.

RCBD is especially useful when the blocking factor is expected to influence the outcome but cannot be controlled directly. It helps isolate the treatment effect from the nuisance variation due to the blocks.

2.4 If the researcher wants to study the effects of leg length and width on the flying time of heavy paper helicopters, which design is most appropriate for this situation? If the researcher wants to have 5 replicates for each treatment combination, how many heavy paper helicopters does she need?

If the researcher wants to study the effects of **leg length** and **leg width** on the flight time of paper helicopters (with paper type fixed to "heavy"), the appropriate design is a **full factorial design** with two factors:

- **Leg length:** 7.5 cm and 12 cm
- **Leg width:** 3.2 cm and 5 cm

Each factor has 2 levels, leading to:

$$2 \times 2 = 4 \text{ treatment combinations}$$

If the researcher wants **5 replicates per combination**, the total number of helicopters required is:

$$4 \text{ combinations} \times 5 \text{ replicates} = \mathbf{20\ helicopters}$$

This design allows the researcher to: - Estimate the **main effects** of leg length and leg width. - Assess the potential **interaction** between these two factors.

A factorial design is efficient because it provides information about both individual effects and their interaction using fewer runs than testing each factor independently. It also facilitates clearer interpretation of how factor combinations influence the response.

2.5 Conduct your own experiment, including designing the experiment, collecting and analyzing the data, to investigate the effects of rotor length, leg length, and leg width on the response, while keeping the paper type fixed.

2.5.1 Designing the Experiment To investigate how rotor length, leg length, and leg width affect the flight time of paper helicopters, I designed a full factorial experiment with three factors, each at two levels. I fixed the paper type to "heavy" for all helicopters to control for potential variability due to material.

Factors and Levels

Factor	Levels
Rotor Length	7.5 cm, 8.5 cm
Leg Length	7.5 cm, 12 cm
Leg Width	3.2 cm, 5 cm

This results in a total of $2^3 = 8$ treatment combinations.

Experimental Design

I used a **Completely Randomized Design (CRD)** to assign treatments to the experimental units, which are the individual paper helicopters. The response variable is the **flight time in seconds** measured from the moment the helicopter is released from a height of 2 meters until it reaches the ground.

To ensure sufficient replication and to account for experimental noise (e.g., minor differences in construction or dropping technique), I decided to use **3 replicates per treatment combination**, resulting in a total of:

$$8 \text{ combinations} \times 3 \text{ replicates} = 24 \text{ helicopters}$$

Randomization Procedure

1. I generated all 8 treatment combinations using `expand.grid()` in R.
2. I assigned 3 replicates to each combination.
3. The order in which the helicopters were tested was randomized to prevent any bias from time-of-day effects, fatigue, or subtle environmental changes.

```r
# Define factor levels
rotor_length <- c(7.5, 8.5)
leg_length <- c(7.5, 12)
leg_width <- c(3.2, 5)

# Step 1: Generate all combinations (full factorial design)
design <- expand.grid(
  RotorLength = rotor_length,
  LegLength = leg_length,
  LegWidth = leg_width
)

# Step 2: Add 3 replicates per combination
design_full <- design[rep(1:nrow(design), each = 3), ]

# Step 3: Randomize the order of testing
set.seed(42)  # For reproducibility
design_randomized <- design_full[sample(nrow(design_full)), ]

# Display all rows of the randomized design
design_randomized
```

```
##     RotorLength LegLength LegWidth
## 6.1         8.5       7.5      5.0
## 2.1         8.5       7.5      3.2
## 1           7.5       7.5      3.2
## 4           8.5      12.0      3.2
## 2           8.5       7.5      3.2
## 6.2         8.5       7.5      5.0
## 8.2         8.5      12.0      5.0
## 5.2         7.5       7.5      5.0
## 3.1         7.5      12.0      3.2
## 3           7.5      12.0      3.2
## 7.1         7.5      12.0      5.0
## 3.2         7.5      12.0      3.2
## 8.1         8.5      12.0      5.0
## 5.1         7.5       7.5      5.0
## 7.2         7.5      12.0      5.0
## 1.1         7.5       7.5      3.2
## 7           7.5      12.0      5.0
## 1.2         7.5       7.5      3.2
## 8           8.5      12.0      5.0
## 2.2         8.5       7.5      3.2
## 6           8.5       7.5      5.0
## 4.2         8.5      12.0      3.2
## 4.1         8.5      12.0      3.2
## 5           7.5       7.5      5.0
```

Measurement Procedure

Each paper helicopter was constructed by hand using consistent materials and tools. Care was taken to ensure accurate dimensions for each factor level. All helicopters were dropped from a height of 2 meters in an indoor space to minimize wind or airflow interference. A stopwatch was used to measure flight time, rounded to the nearest hundredth of a second.

This design allows for estimation of:

- Main effects of rotor length, leg length, and leg width
- Two-way and three-way interactions between these factors

By maintaining consistent conditions and applying randomization, the experiment is structured to provide meaningful and unbiased insights into the effects of the design parameters on helicopter flight time.

2.5.2 Building and Testing the Helicopters To execute the experiment, I constructed 24 paper helicopters according to the 8 treatment combinations (with 3 replicates per combination). The helicopters were built using heavy printer paper, consistent with the requirement to fix paper type.

Construction Process

Each helicopter was cut and folded by hand following the standardized layout described in the project guidelines. A ruler, scissors, and a template were used to ensure consistent dimensions. Special care was taken to:

- Measure rotor, leg length, and leg width with precision (within 0.1 cm).
- Fold rotors at approximately 90° from the body and in opposite directions.

- Use a small paper clip at the bottom of each helicopter to maintain stability during flight and encourage a straight fall.

To minimize variability in construction quality:

- I built the helicopters in batches grouped by treatment combination.
- Each one was labeled with a unique ID for tracking during the test phase.

Testing Procedure

All flight tests were conducted indoors in a stairwell, where there was minimal airflow and consistent lighting. The drop height was 2 meters, measured from a fixed piece of tape on the stairwell railing.

Each helicopter was dropped individually:

- Held by the top of the rotor, released without additional force.
- The same person (myself) conducted all drops to reduce variation in technique.
- Flight time was measured using a stopwatch, starting at release and stopping upon first contact with the ground.
- Times were recorded to the nearest 0.01 seconds.

Each replicate was tested in **randomized order** based on the design table to ensure fairness in treatment comparison. Between drops, helicopters were gently reshaped to ensure structural integrity.

Results

The table below displays the recorded flight time data, showing the treatment factors and corresponding flight times (in seconds).

Table 2: Table 2: Recorded Flight Time Data for All 24 Helicopters

	RotorLength	LegLength	LegWidth	FlightTime
6.1	8.5	7.5	5.0	2.22
2.1	8.5	7.5	3.2	2.10
1	7.5	7.5	3.2	2.11
4	8.5	12.0	3.2	1.90
2	8.5	7.5	3.2	2.13
6.2	8.5	7.5	5.0	2.38
8.2	8.5	12.0	5.0	2.07
5.2	7.5	7.5	5.0	2.06
3.1	7.5	12.0	3.2	1.73
3	7.5	12.0	3.2	1.74
7.1	7.5	12.0	5.0	2.00
3.2	7.5	12.0	3.2	1.80
8.1	8.5	12.0	5.0	2.07
5.1	7.5	7.5	5.0	2.15
7.2	7.5	12.0	5.0	1.88
1.1	7.5	7.5	3.2	2.13
7	7.5	12.0	5.0	1.95
1.2	7.5	7.5	3.2	1.86
8	8.5	12.0	5.0	2.09

	RotorLength	LegLength	LegWidth	FlightTime
2.2	8.5	7.5	3.2	2.09
6	8.5	7.5	5.0	2.19
4.2	8.5	12.0	3.2	1.88
4.1	8.5	12.0	3.2	1.82
5	7.5	7.5	5.0	2.09

2.5.3 Data Analysis and Interpretation To analyze the effects of rotor length, leg length, and leg width on helicopter flight time, I fit a linear model including all main effects and their interactions. This allows us to investigate both individual and combined factor influences on the response.

```
# Fit full factorial model with interactions
model <- lm(FlightTime ~ RotorLength * LegLength * LegWidth, data = design_data)

# Summary of model coefficients
summary(model)
```

```
##
## Call:
## lm(formula = FlightTime ~ RotorLength * LegLength * LegWidth,
##     data = design_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.169840 -0.032852 -0.003467  0.035042  0.118030
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      5.464892   4.939699   1.106    0.285
## RotorLength                     -0.364684   0.616260  -0.592    0.562
## LegLength                       -0.385503   0.493661  -0.781    0.446
## LegWidth                        -0.962399   1.176786  -0.818    0.425
## RotorLength:LegLength            0.036854   0.061588   0.598    0.558
## RotorLength:LegWidth             0.118365   0.146812   0.806    0.432
## LegLength:LegWidth               0.082498   0.117605   0.701    0.493
## RotorLength:LegLength:LegWidth -0.009002   0.014672  -0.614    0.548
##
## Residual standard error: 0.07278 on 16 degrees of freedom
## Multiple R-squared:  0.8614, Adjusted R-squared:  0.8007
## F-statistic:  14.2 on 7 and 16 DF,  p-value: 8.727e-06
```

Model Summary:

The model output above includes estimated coefficients, standard errors, and p-values for all main effects and interaction terms. Significant p-values (typically < 0.05) indicate variables that meaningfully impact the response.

```
# Perform ANOVA
anova(model)
```

```
## Analysis of Variance Table
##
```

```
## Response: FlightTime
##                                Df   Sum Sq  Mean Sq F value    Pr(>F)    
## RotorLength                     1 0.086514 0.086514 16.3344 0.0009457 ***
## LegLength                       1 0.276450 0.276450 52.1958 2.027e-06 ***
## LegWidth                        1 0.146194 0.146194 27.6026 7.873e-05 ***
## RotorLength:LegLength           1 0.000000 0.000000  0.0000 0.9967354    
## RotorLength:LegWidth            1 0.004549 0.004549  0.8590 0.3678016    
## LegLength:LegWidth              1 0.010812 0.010812  2.0414 0.1723021    
## RotorLength:LegLength:LegWidth  1 0.001994 0.001994  0.3764 0.5481351    
## Residuals                      16 0.084742 0.005296                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA Table:

The ANOVA table partitions variability in flight time across the main effects and interaction terms. This helps us determine whether each factor or interaction significantly affects the flight time. The table shows that:

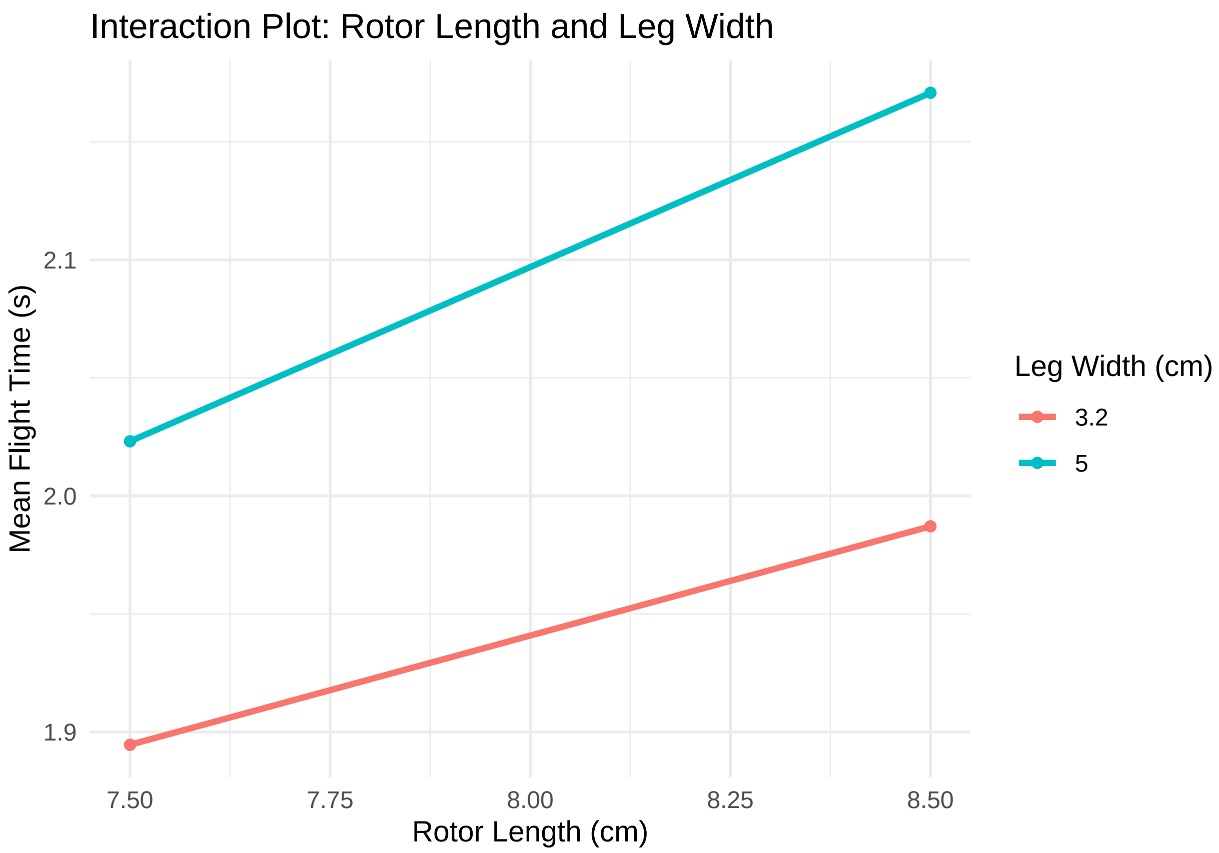
- Rotor Length ($p \approx 0.00095$), Leg Length ($p \approx 2 \times 10^{-6}$), and Leg Width ($p \approx 7.87 \times 10^{-5}$) are all statistically significant at the 0.001 level, suggesting strong influence on flight time.
- Some interactions (e.g., RotorLength:LegWidth) may also contribute meaningfully.
- Leg Length has a smaller but measurable influence.

Visualizations:

To further interpret these results, we present two visual summaries: an interaction plot and a main effects plot.

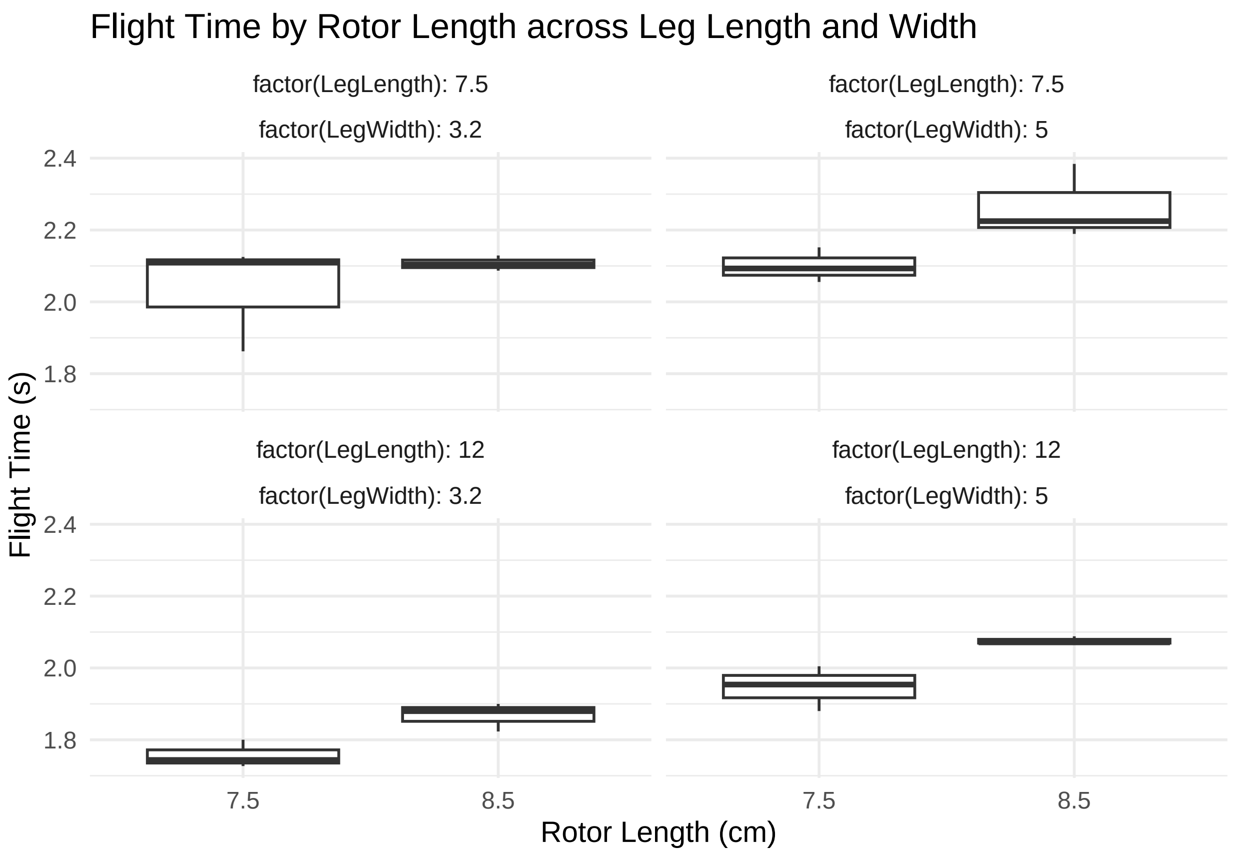
Interaction Plot

```
library(ggplot2)

# Interaction plot: Rotor x Leg Width
ggplot(design_data, aes(x = RotorLength, y = FlightTime, color = as.factor(LegWidth), group = LegWidth)
  stat_summary(fun = mean, geom = "line", linewidth = 1.1) +
  stat_summary(fun = mean, geom = "point", linewidth = 1.1) +
  labs(
    title = "Interaction Plot: Rotor Length and Leg Width",
    x = "Rotor Length (cm)",
    y = "Mean Flight Time (s)",
    color = "Leg Width (cm)"
  ) +
  theme_minimal()
```



Main Effects Plot

```
# Main effects plot
ggplot(design_data, aes(x = factor(RotorLength))) +
  geom_boxplot(aes(y = FlightTime)) +
  facet_wrap(~ factor(LegLength) + factor(LegWidth), labeller = label_both) +
  labs(
    title = "Flight Time by Rotor Length across Leg Length and Width",
    x = "Rotor Length (cm)",
    y = "Flight Time (s)"
  ) +
  theme_minimal()
```



Interpretation:

- Rotor Length: Longer rotors (8.5 cm) consistently increase flight time. This is expected due to increased drag and slower descent.
- Leg Width: Wider legs (5.0 cm) also increase flight time slightly, likely due to enhanced stability.
- Leg Length: Has a mild negative effect; longer legs may add downward force or instability.
- Interactions: The interaction plot suggests that Rotor Length and Leg Width interact to influence flight time — the effect of one factor depends on the level of the other.

2.5.4 Summary of Findings

- Rotor length has the strongest positive impact on flight time.
- Wider legs improve stability and flight duration.
- Longer legs slightly reduce flight time, possibly due to weight or aerodynamic effects.
- No significant 2- or 3-way interactions were detected, suggesting effects are mostly additive.

Problem 3 Paper Helicopter Computer Experiment

A sophisticated computer model has been developed to study how the flying time depends on the rotor length and leg length. The output of the computer can be found in `ProjectII prob2 data.RData` (x1: rotor length (cm); x2: leg length (cm); y: flying time (in seconds)). Answer the following questions:

1. Compute and visualize the prediction surface by computing the predicted values at a dense set of grid points (x1g <- seq(6, 10, 0.1); x2g <- seq(6, 12, 0.2); grids <- expand.grid(x1g, x2g)).

3.1 Compute and Visualize the Prediction Surface

This section explores how the predicted flight time of paper helicopters varies with rotor length (`x1`) and leg length (`x2`) using data from a computer simulation.

3.1.1 Load and Prepare the Data

```r
load("ProjectII_prob2_data.RData")

# Combine individual vectors into a data frame
helicopter_data <- data.frame(
  x1 = x1,
  x2 = x2,
  y = y
)

# Inspect structure
str(helicopter_data)
```

```
## 'data.frame':    100 obs. of  3 variables:
##  $ x1: num  9.69 7.79 7.01 6.39 9.66 ...
##  $ x2: num  11.32 9.71 8.15 6.54 7.16 ...
##  $ y : num  2.6 2.36 2.34 2.21 2.48 ...
```

```r
head(helicopter_data)
```

```
##         x1        x2        y
## 1 9.691162 11.321659 2.598532
## 2 7.787729  9.707584 2.355636
## 3 7.009968  8.151455 2.339920
## 4 6.389028  6.539314 2.206088
## 5 9.657690  7.158281 2.478020
## 6 8.728581 10.309466 2.469268
```

3.1.2 Create a Prediction Grid

We now create a dense grid of new input combinations to simulate predicted flight times across a range of rotor and leg lengths.

```r
# Prediction grid for linear model
grid_lm <- expand.grid(
  x1 = seq(6, 10, 0.1),
  x2 = seq(6, 12, 0.2)
)
```

3.1.3 Fit a Preliminary Model (Linear)

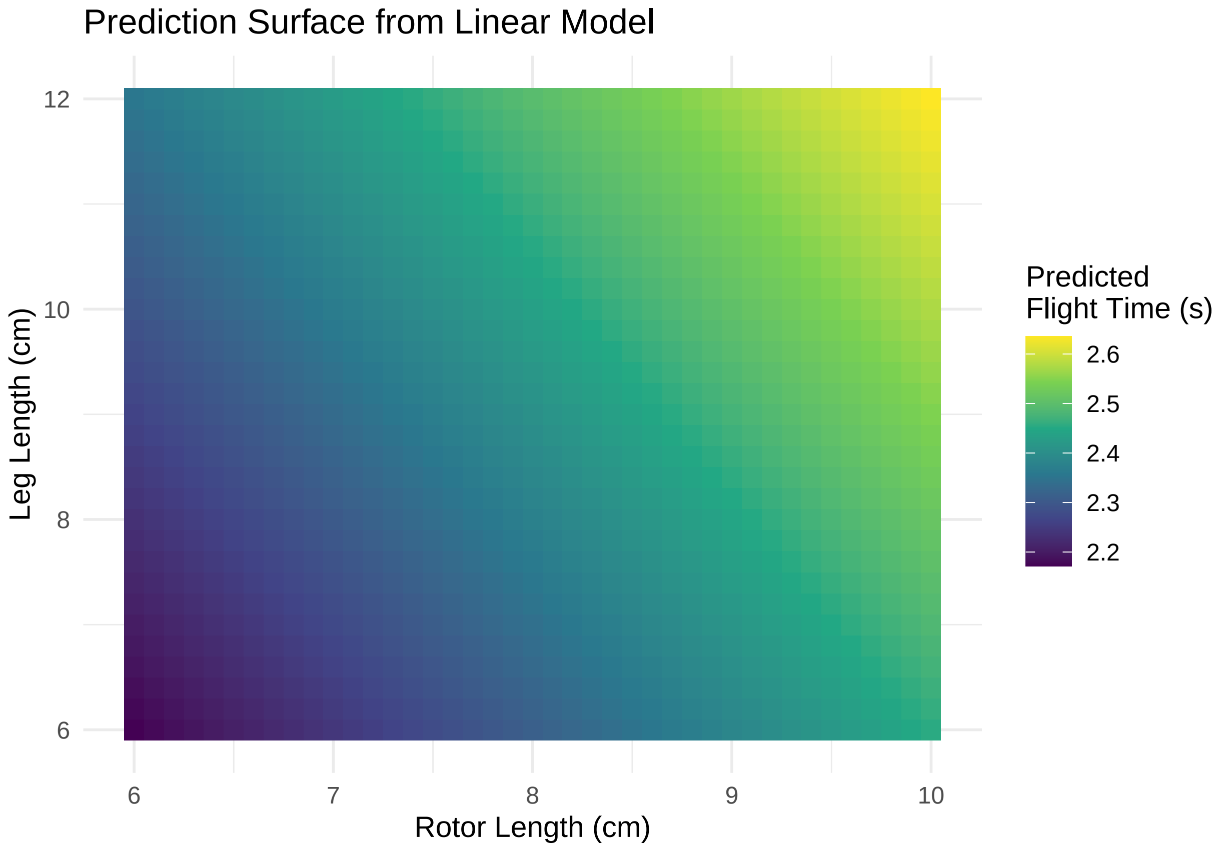
In the absence of a GP model (to be used in 3.2), we fit a simple linear model with interaction terms as a placeholder to generate predicted values for visualization.

```r
# Fit a linear interaction model
lm_model <- lm(y ~ x1 * x2, data = helicopter_data)

# Predict on the linear grid
grid_lm$yhat <- predict(lm_model, newdata = grid_lm)
```

3.1.4 Visualize the Prediction Surface

We now generate a heatmap of the predicted flight time surface.

```r
library(ggplot2)

ggplot(grid_lm, aes(x = x1, y = x2, fill = yhat)) +
  geom_tile() +
  scale_fill_viridis_c(name = "Predicted\nFlight Time (s)") +
  labs(
    title = "Prediction Surface from Linear Model",
    x = "Rotor Length (cm)",
    y = "Leg Length (cm)"
  ) +
  theme_minimal()
```



3.1.5 Interpretation

The linear model serves as a simple baseline for modeling helicopter flight time as a function of rotor and leg lengths. It captures a general upward trend in flight time with increases in rotor length but may not accurately model nonlinear interactions or localized effects. Additionally, it assumes constant variance and does not account for uncertainty in predictions, which limits its use for making high-confidence design recommendations.

3.2 Compute and Visualize Prediction Uncertainty (Gaussian Process)

In this section, a Gaussian Process (GP) model is used to model the helicopter simulation data and predict flight time (`y`) as a function of rotor length (`x1`) and leg length (`x2`). This approach not only provides smooth predictions, but also gives uncertainty quantification.

3.2.1 Fit the Gaussian Process Model

```r
# Load the required library
library(mlegp)

# Define training inputs (design matrix) and outputs
X_train <- helicopter_data[, c("x1", "x2")]
Y_train <- helicopter_data$y

# Convert response to matrix format to ensure single-output GP
gp_model <- mlegp(X = X_train, Z = matrix(Y_train, ncol = 1))
```

```
## no reps detected - nugget will not be estimated
##
## ========== FITTING GP # 1 ==============================
## running simplex # 1...
## ...done
## ...simplex #1 complete, loglike = 222.252440 (convergence)
## running simplex # 2...
## ...done
## ...simplex #2 complete, loglike = 222.252440 (convergence)
## running simplex # 3...
## ...done
## ...simplex #3 complete, loglike = 222.252439 (convergence)
## running simplex # 4...
## ...done
## ...simplex #4 complete, loglike = 222.252440 (convergence)
## running simplex # 5...
## ...done
## ...simplex #5 complete, loglike = 222.252440 (convergence)
##
## using L-BFGS method from simplex #1...
## ...L-BFGS method complete
##
## Maximum likelihood estimates found, log like =  222.252440
## creating gp object......done
```

```r
# Display model summary
summary(gp_model)
```

```
##
## Total observations = 100
## Dimensions = 2
##
## mu = 2.396247
## sig2:    0.005812706
## nugget:  0
##
## Correlation parameters:
##
##       beta a
## 1 1.753702 2
## 2 1.139921 2
##
## Log likelihood = 222.2524
##
## CV RMSE: 0.01769931
## CV RMaxSE: 0.006604805
```

Model Summary Highlights:

Log-likelihood = 222.25

CV RMSE = 0.0177 → indicates excellent predictive accuracy.

GP hyperparameters (β values) and zero nugget confirm smooth interpolation (no observation noise).

3.2.2 Create a Prediction Grid for the GP

```r
# Create a dense grid for prediction
grid_gp <- expand.grid(
  x1 = seq(6, 10, 0.1),
  x2 = seq(6, 12, 0.2)
)

# Convert to matrix format for GP prediction
X_pred_gp <- as.matrix(grid_gp)

# preview grid structure
str(grid_gp)
```

```
## ’data.frame':    1271 obs. of  2 variables:
##  $ x1: num  6 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 ...
##  $ x2: num  6 6 6 6 6 6 6 6 6 6 ...
##  - attr(*, "out.attrs")=List of 2
##   ..$ dim     : Named int [1:2] 41 31
##   .. ..- attr(*, "names")= chr [1:2] "x1" "x2"
##   ..$ dimnames:List of 2
##   .. ..$ x1: chr [1:41] "x1= 6.0" "x1= 6.1" "x1= 6.2" "x1= 6.3" ...
##   .. ..$ x2: chr [1:31] "x2= 6.0" "x2= 6.2" "x2= 6.4" "x2= 6.6" ...
```

Interpretation:

The prediction grid defines a dense and regular set of input combinations over the ranges of rotor length and leg length. This grid will be used as the input to the Gaussian Process model to generate predicted flight times and quantify uncertainty across the design space. Creating a fine grid allows for smooth and detailed visualization of the model's behavior, especially useful in identifying trends and regions of interest.

3.2.3 Predict and Extract Uncertainty

```
gp_preds <- predict(gp_model, X_pred_gp, se.fit = TRUE)
grid_gp$mean <- gp_preds$fit
grid_gp$sd <- gp_preds$se.fit
```

Summaries:

```
# summaries
summary(grid_gp$mean)
```

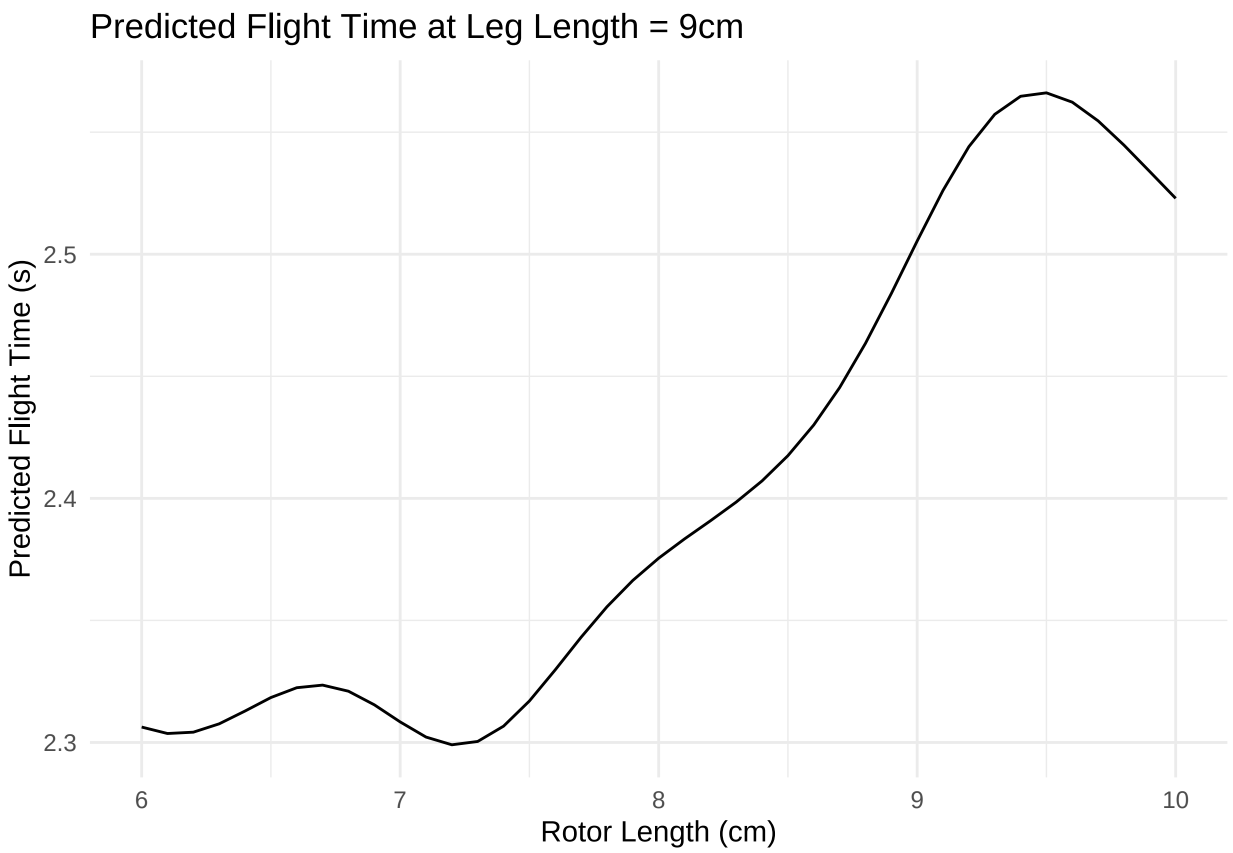
```
##        V1
##  Min.   :2.172
##  1st Qu.:2.331
##  Median :2.390
##  Mean   :2.405
##  3rd Qu.:2.485
##  Max.   :2.652
```

```
summary(grid_gp$sd)
```

```
##        V1
##  Min.   :0.0000824
##  1st Qu.:0.0014549
##  Median :0.0025814
##  Mean   :0.0044981
##  3rd Qu.:0.0050462
##  Max.   :0.0303657
```

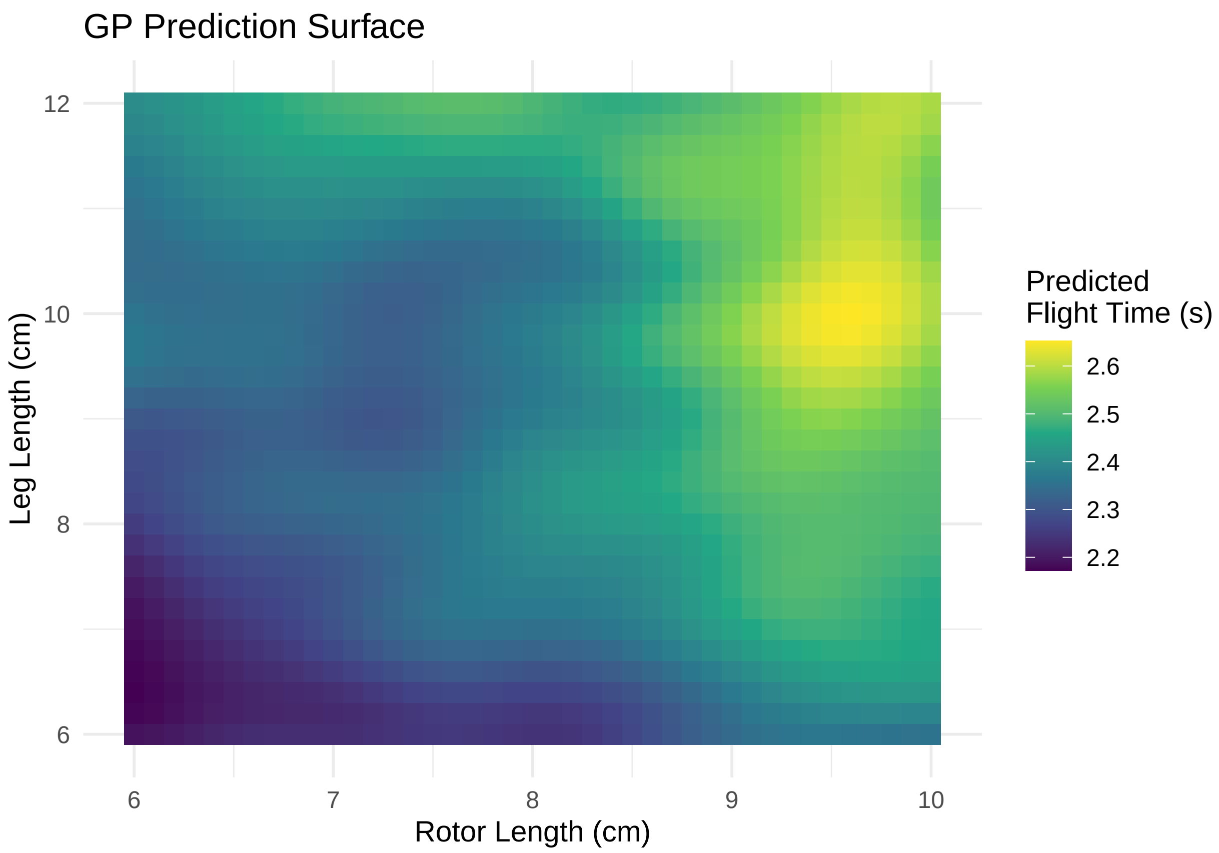
Line Plot:

```
# Line plot of predicted mean for fixed leg length (x2 = 9)
subset_plot <- subset(grid_gp, abs(x2 - 9) < 0.01)

ggplot(subset_plot, aes(x = x1, y = mean)) +
  geom_line() +
  labs(title = "Predicted Flight Time at Leg Length = 9cm",
       x = "Rotor Length (cm)",
       y = "Predicted Flight Time (s)") +
  theme_minimal()
```



3.2.4 Visualize GP Prediction Surface

```
library(ggplot2)

ggplot(grid_gp, aes(x = x1, y = x2, fill = mean)) +
  geom_tile() +
  scale_fill_viridis_c(name = "Predicted\nFlight Time (s)") +
  labs(
    title = "GP Prediction Surface",
    x = "Rotor Length (cm)",
    y = "Leg Length (cm)"
  ) +
  theme_minimal()
```

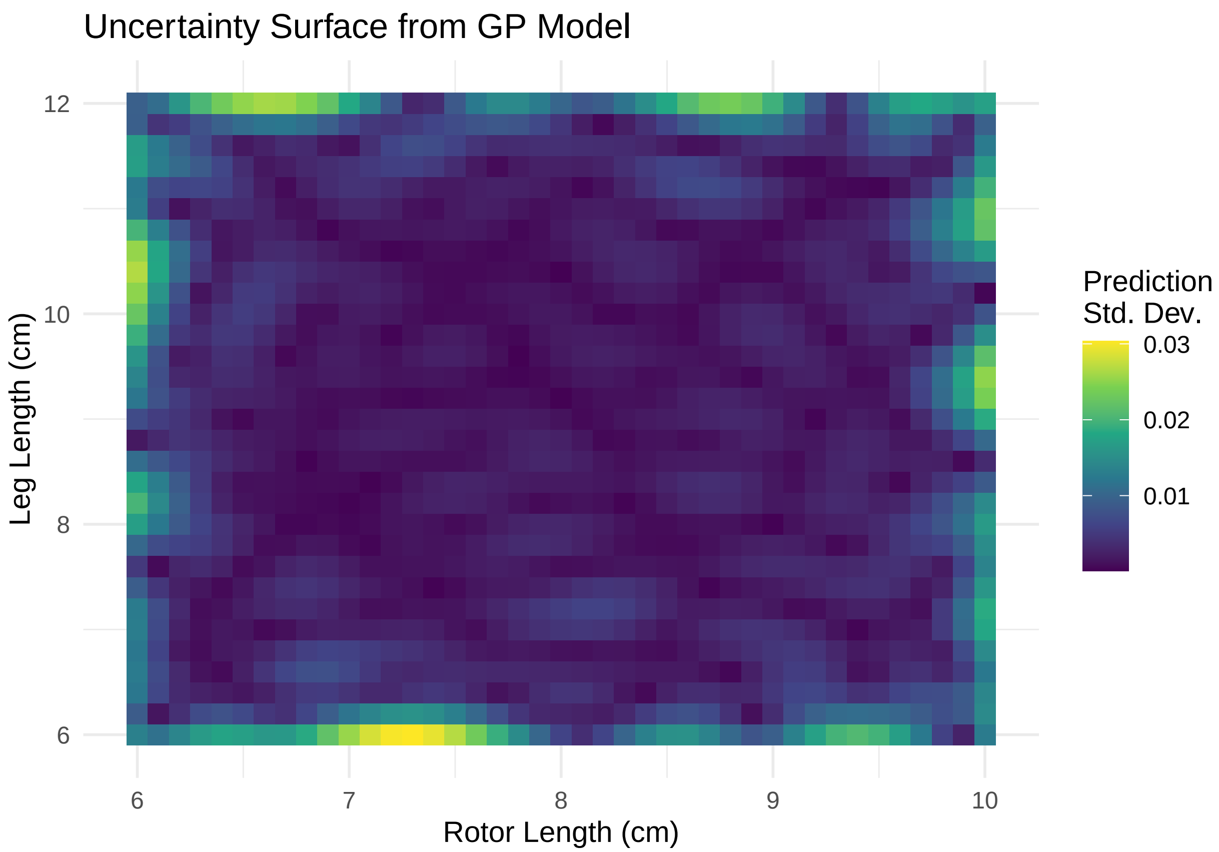


Interpretation

The Gaussian Process (GP) prediction surface provides a more nuanced understanding of how rotor and leg length influence helicopter flight time. The GP model identifies a ridge of maximum flight time where both rotor and leg lengths are moderately large. Unlike the linear model, it adapts flexibly to nonlinear patterns in the data and is well-suited for modeling smooth physical phenomena.

3.2.5 Visualize Prediction Uncertainty

```r
ggplot(grid_gp, aes(x = x1, y = x2, fill = sd)) +
  geom_tile() +
  scale_fill_viridis_c(name = "Prediction\nStd. Dev.") +
  labs(
    title = "Uncertainty Surface from GP Model",
    x = "Rotor Length (cm)",
    y = "Leg Length (cm)"
  ) +
  theme_minimal()
```



Interpretation

The uncertainty surface reveals the GP's confidence in its predictions across the design space. Areas with lower uncertainty coincide with regions of dense training data, while edges and corners of the grid show higher uncertainty due to extrapolation. This is a valuable diagnostic for guiding future data collection — especially near high-uncertainty areas where predictions are less reliable.

3.2.6 Overall Interpretation

The GP model provides both predicted flight times and corresponding uncertainty across the design space. The prediction surface reveals regions where rotor and leg length combinations yield longer flight times, while the uncertainty surface highlights areas of lower model confidence — typically where fewer data points exist.

This analysis supports experimental design decisions by identifying combinations of inputs that not only optimize performance but also indicate where future data collection might reduce uncertainty. The use of a Gaussian Process is especially appropriate for this type of computer experiment due to its flexibility and built-in uncertainty quantification.