Assignment 9

Decision analysis

anonymous

1 General information

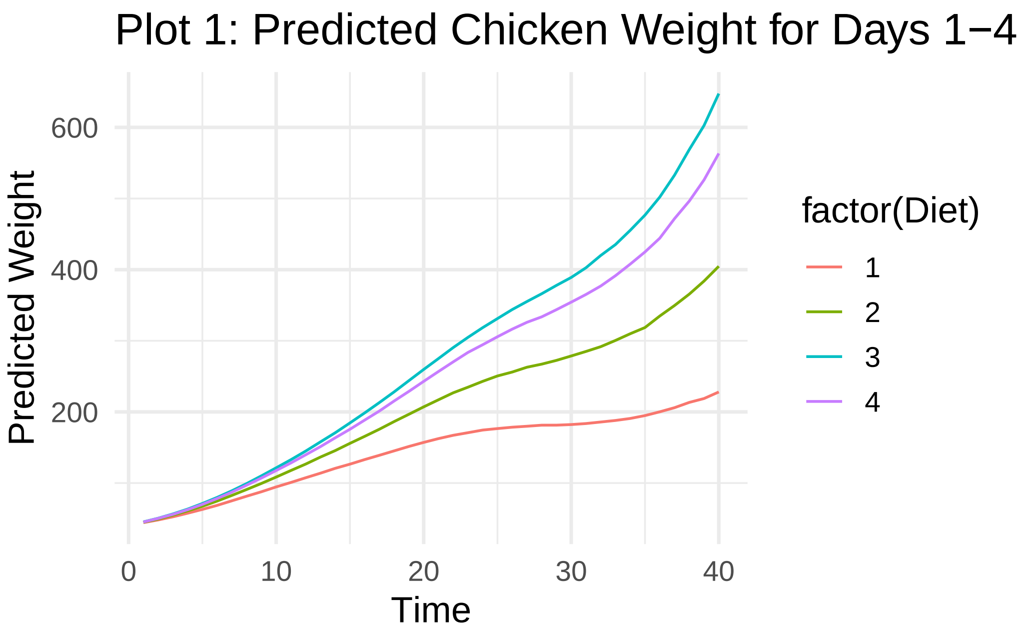
This is the template for assignment 9. You can download the qmd-file or copy the code from this rendered document after clicking on `</> Code` in the top right corner.

Please replace the instructions in this template by your own text, explaining what you are doing in each exercise.

2 Escaping from the chicken coop

2.1 (a)

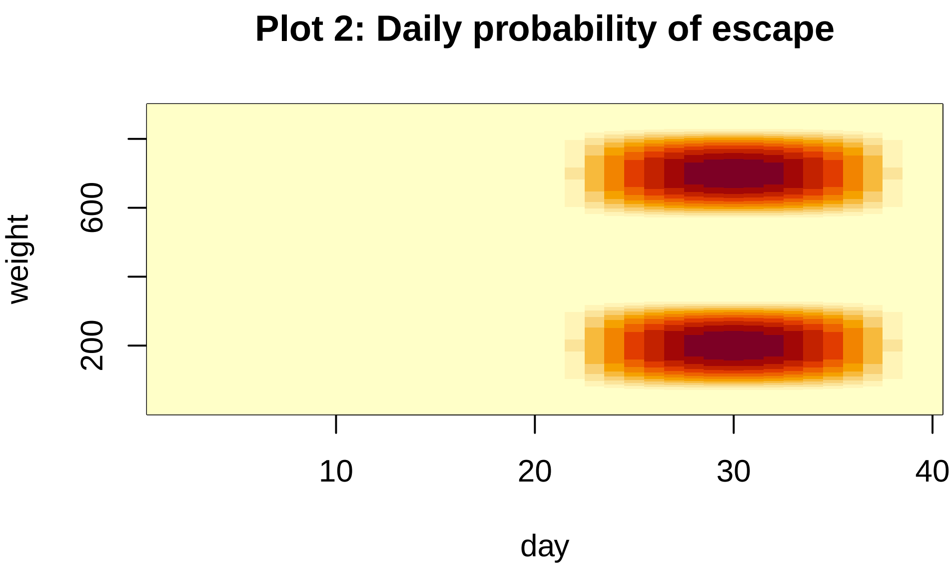
```
# Creating new data for prediction
new_data <- expand.grid(Time = 1:40, Diet = unique(ChickWeight$Diet))

# Generating predictions
ppe <- posterior_predict(fit, newdata=new_data, allow_new_levels=TRUE, sample_new_levels="gaussian")




# Assuming 'Category' is the column to split by and it has 4 unique values (1, 2, 3, 4)
# Splitting the data frame

ppe_1 = ppe[,1:40]
ppe_2 = ppe[,41:80]
ppe_3 = ppe[,81:120]
ppe_4 = ppe[,120:160]


# Computing means for each combination of Time and Diet
ppe_means <- colMeans(ppe)
new_data$predicted_weight <- ppe_means


# Visualization
ggplot(new_data, aes(x=Time, y=predicted_weight, color=factor(Diet))) +
  geom_line() +
  labs(title="Plot 1: Predicted Chicken Weight for Days 1-40 per Diet", y="Predicted Weight")
```



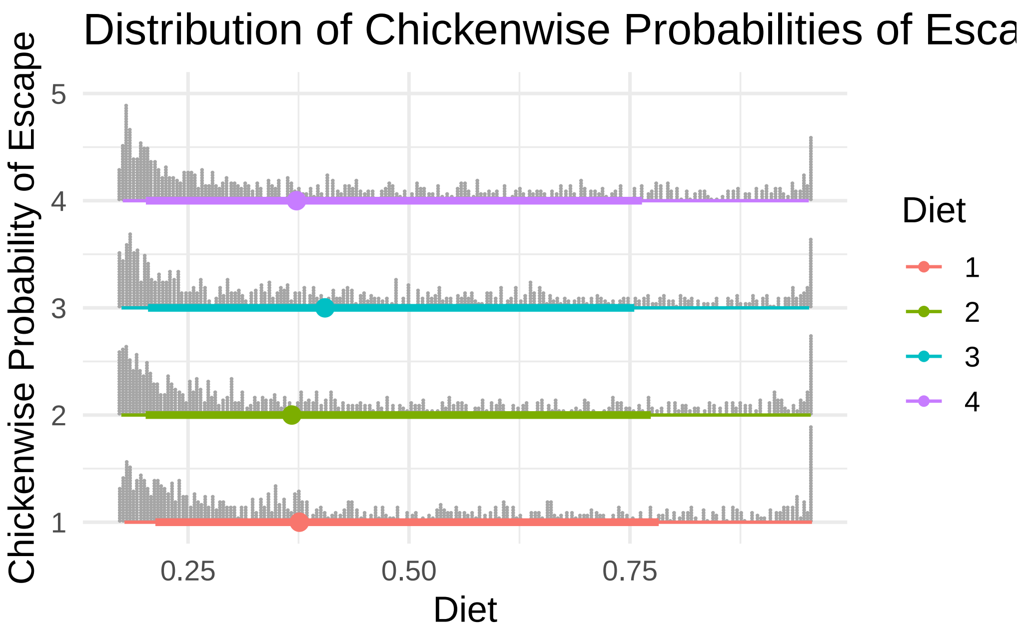
The predictions look reasonable as they follow the same trend as in last weeks assignment. In this the order of superiority for diet was also diet 3 > diet 4 > diet 2 > diet 1. Although the trend is different for the data (before day 21) where the trend seems to be linear increase and the predicted values (including all values after day 21) seem to follow polynomial or exponetial increase.

2.2 (b)

```
# Useful r functions: chickenwise_probability_of_escape (see above)
# rep(..., each=...), apply,
# ggplot, stat_dotsinterval

days = 1:40
weights = 1:900
heatmap_matrix = outer(days,weights,daily_probability_of_escape)
image(days, weights, heatmap_matrix, xlab="day", ylab="weight", main="Plot 2: Daily probability of e
```




```r
chickenwise_prob_escape <- function(weights) {
  prob_escape = chickenwise_probability_of_escape(weights)
  return(1 - prob_escape)
}

# Compute chickenwise probabilities for each set of predictions

chickenwise_probs <- apply(ppe, 1, chickenwise_prob_escape)

# Create a data frame for diet-wise probabilities
dietwise_probs <- data.frame(Diet = rep(1:4, each = ncol(ppe)/4),
                              Chickenwise_Prob = chickenwise_probs)

# Plot the distribution of chickenwise probabilities per diet with uncertainty intervals
ggplot(dietwise_probs, aes(x = Chickenwise_Prob  , y = (Diet), group = Diet, color = as.factor(Diet)
  stat_dotsinterval(dotsize = 0.8) +
  labs(title = "Distribution of Chickenwise Probabilities of Escape per Diet",
       x = "Diet",
       y = "Chickenwise Probability of Escape") +
  scale_color_discrete(name = "Diet")
```



2.3 (c)

```r
# Useful r functions: chickenwise_probability_of_escape (see above)
# apply, aggregate,
expected_escape_by_diet <- aggregate(Chickenwise_Prob ~ Diet ,data = dietwise_probs, mean)
expected_escape_by_diet
```

```
  Diet Chickenwise_Prob
1    1        0.4634452
2    2        0.4533953
3    3        0.4634716
4    4        0.4531872
```

```r
#wrong_probaility <- apply(new_data$predicted_weight, 1, chickenwise_prob_escape)

a_1<- filter(new_data, Diet == 1)
a_2<- filter(new_data, Diet == 2)
a_3<- filter(new_data, Diet == 3)
a_4<- filter(new_data, Diet == 4)

a_1$Diet<- NULL
a_2$Diet<- NULL
a_3$Diet<- NULL
a_4$Diet<- NULL

apply(a_1, 2, chickenwise_probability_of_escape)
```

```
            Time predicted_weight
      0.04348514       0.82715271
```

```r
apply(a_2, 2, chickenwise_probability_of_escape)
```

```
       Time predicted_weight
 0.04348514       0.65659555
```

```r
apply(a_3, 2, chickenwise_probability_of_escape)
```

```
       Time predicted_weight
 0.04348514       0.09461540
```

```r
apply(a_4, 2, chickenwise_probability_of_escape)
```

```
       Time predicted_weight
 0.04348514       0.19270394
```

The reason for not using chickenwise__probability__of__escape * daily expected chicken weights per diet is that these numbers are not comparable, as in this instance we would be multiplying a expected value with individual chicken weight for each day. For example some chickens will be larger and some smaller, these will have different growth speeds which will not always be linear. This information is lost when reducing the dimensions by taking the expected value.

Correct values

Diet Prob

1 0.4583614
2 0.4363192
3 0.4494797
4 0.4464276

Wrong values

Diet Prob

1 0.82516086
2 0.68354484
3 0.12826445
4 0.18574863

The correct values have similar probabilities for all chicken diets with values howering around 44%-46%. However, the incorrect values have significantly higher variance with value range from 13%-83%. The other interesting part is that the wrong ranges are neigher higher or lower but both.

The reason is quite simple and best understood by analysing plot 1 and 2. For diet 1, values nearly perfectly align when its probable for chickens to escape (days=30), this assumes all chickens are the perfect weight to escape although the truth is far from this. The assumption error one makes is that the mean is the true value and the whole distribution is only centered at the mean value for each timeindex. The same can be seen with diet 3 which has a very low probablity with wrong values. The probability depends on individual chickens weigth and time it is this weight.

#AI was my friend as it helped me with interpetaions and R syntax