Animal Information Dataset
Data Cleaning, Analysis, and Predictive Modeling

Burak ER

Introduction

- Objective: Clean, Analyze, and make Predictive model to the Animal Dataset
- Tools Used: Rstudio and R language
- Dataset: 'Animal Dataset.csv'

Data Cleaning Procedures

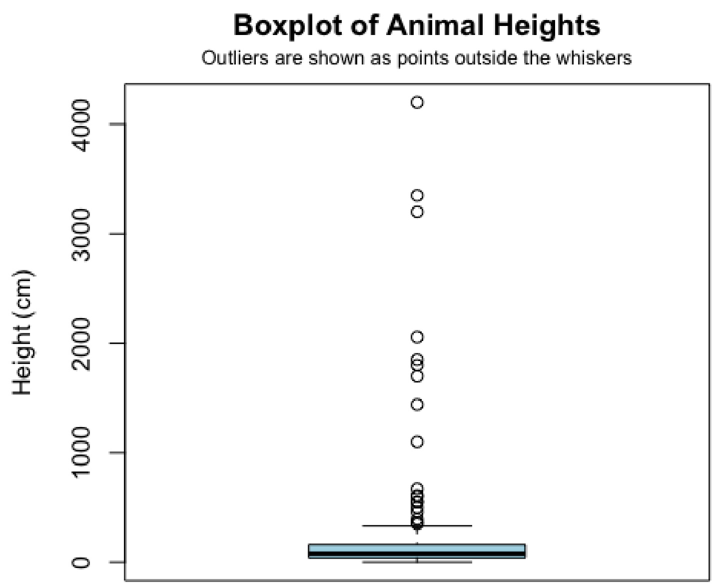
- - Handling 'Up to' phrases
- - Averaging ranges
- - Unit conversions
- - Replacing non-numeric entries
- - Saving cleaned dataset

Basking Shark	Up to 1100	400–700
Bearded Dragon	Up to 60	Up to 600
Bengal Fox	35–40	2.5–4
Bengal Tiger	90–110	220–260
Black Rhinoceros	132–180	800–1,400
Blobfish	Up to 30	Up to 10
Blobfish	Not Applicable	Not Applicable

Basking Shark	1100.00	5.500e+02
Bearded Dragon	60.00	6.000e+02
Bengal Fox	37.50	3.250e+00
Bengal Tiger	100.00	2.400e+02
Black Rhinoceros	156.00	1.100e+03
Blobfish	30.00	1.000e+01
Blobfish	NA	NA

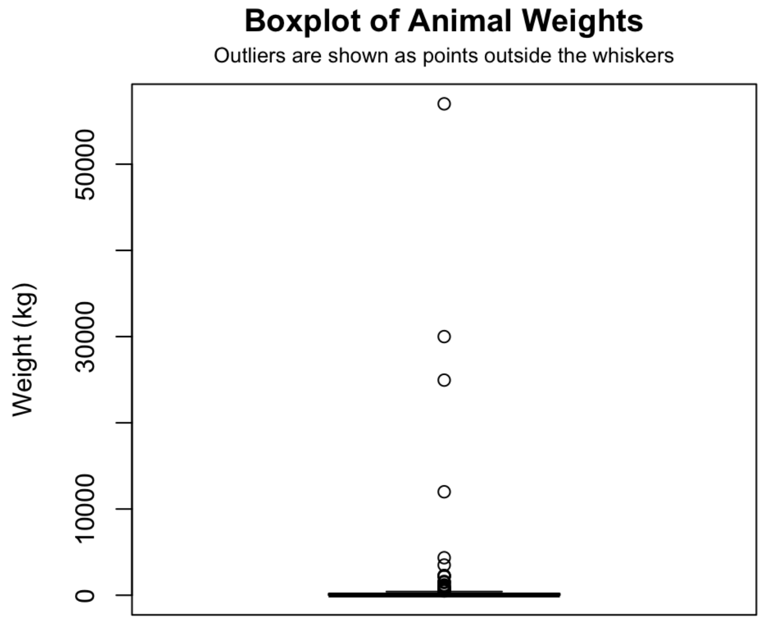
Outlier Analysis: Height

- Woolly Mammoth (42m)
- Blue Whale (32m) identified as outliers.



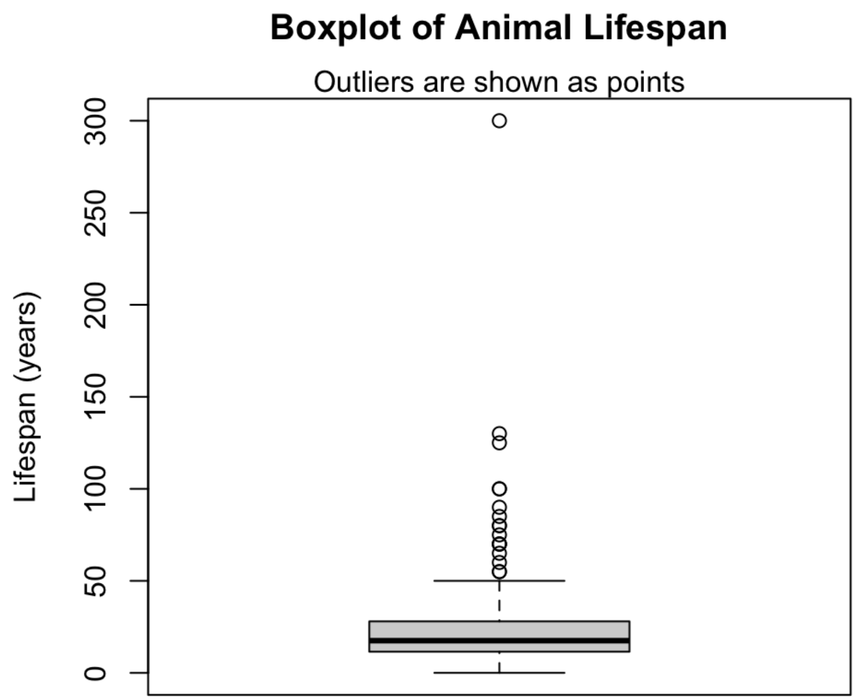
Outlier Analysis: Weight

- Sperm Whale (57 tons)
- Humpback Whale (30 tons) are the examples of outliers.



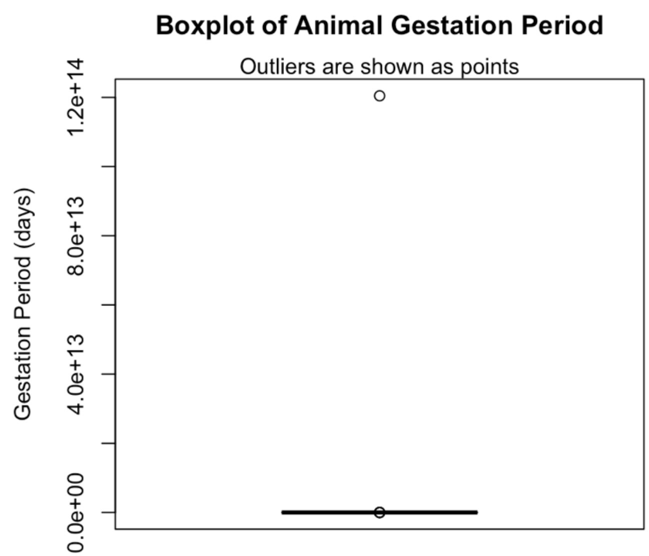
Outlier Analysis: Lifespan

- Hagfish lives up to 300 years, a remarkable outlier.



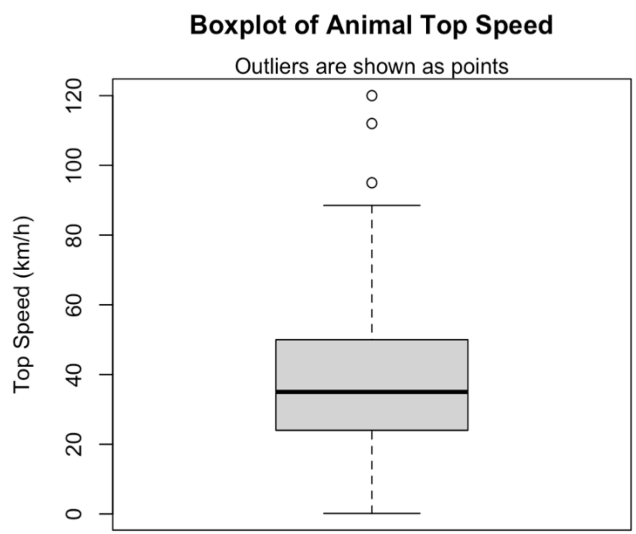
Outlier Analysis: Gestation Period

- Coelacanth shows an extreme exceptional gestation period (~295 million years).
- Basking Shark and African Elephant: 650 days.



Outlier Analysis: Speed Metrics

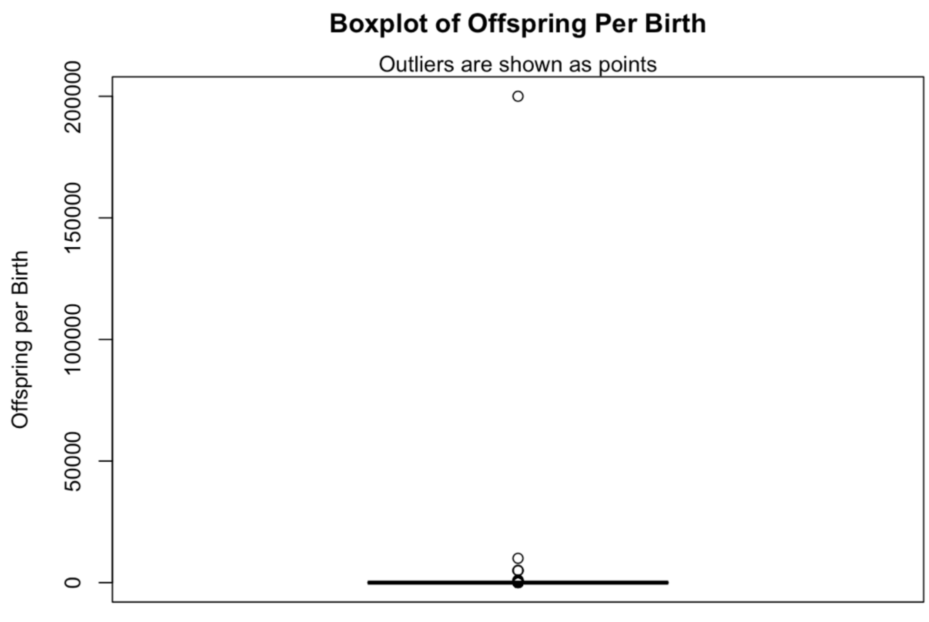
- Bald Eagle (120 km/h)
- Cheetah (112 km/h)
- There are no outliers that are significantly differs in this aspect.



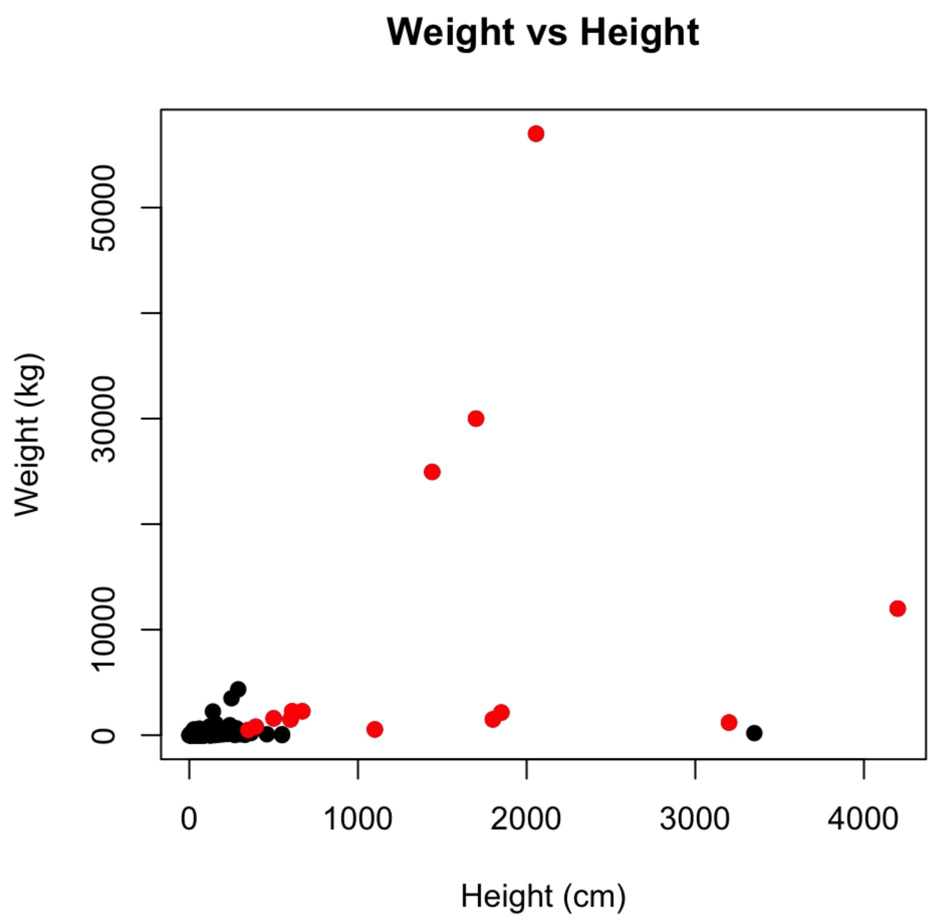
Outlier Analysis: Offspring per Birth

- Giant Pacific Octopus: 200,000 offspring.
- Peacock Mantis Shrimp: 10,000 offspring.

- Remarkeble outliers

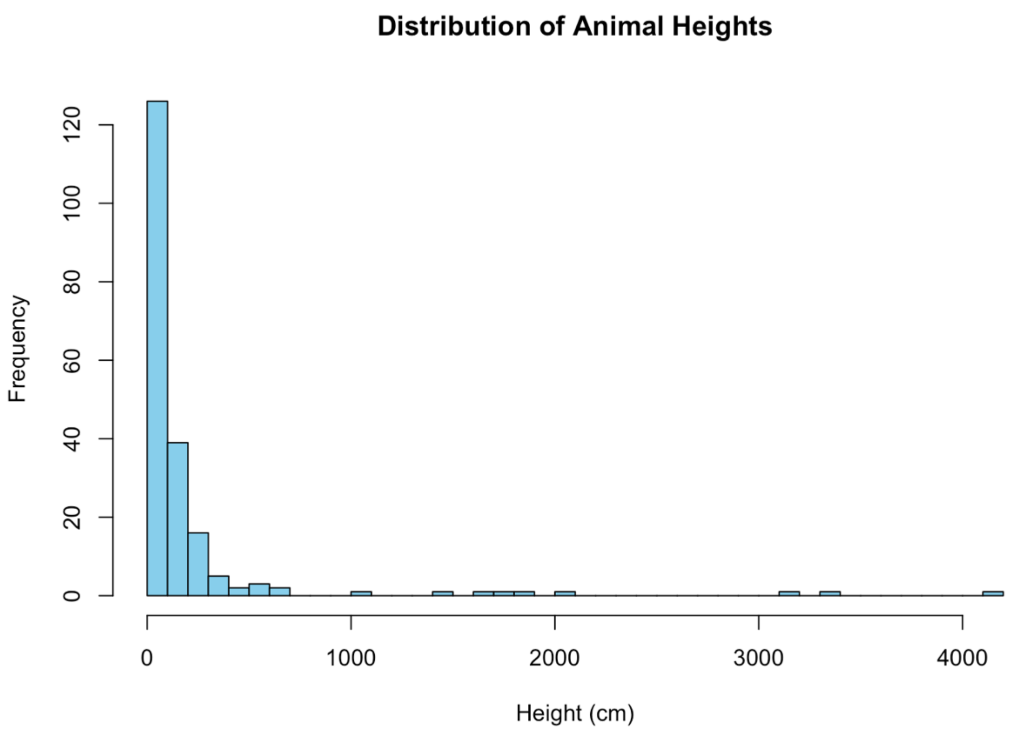


Outlier Analysis: Weight, Height



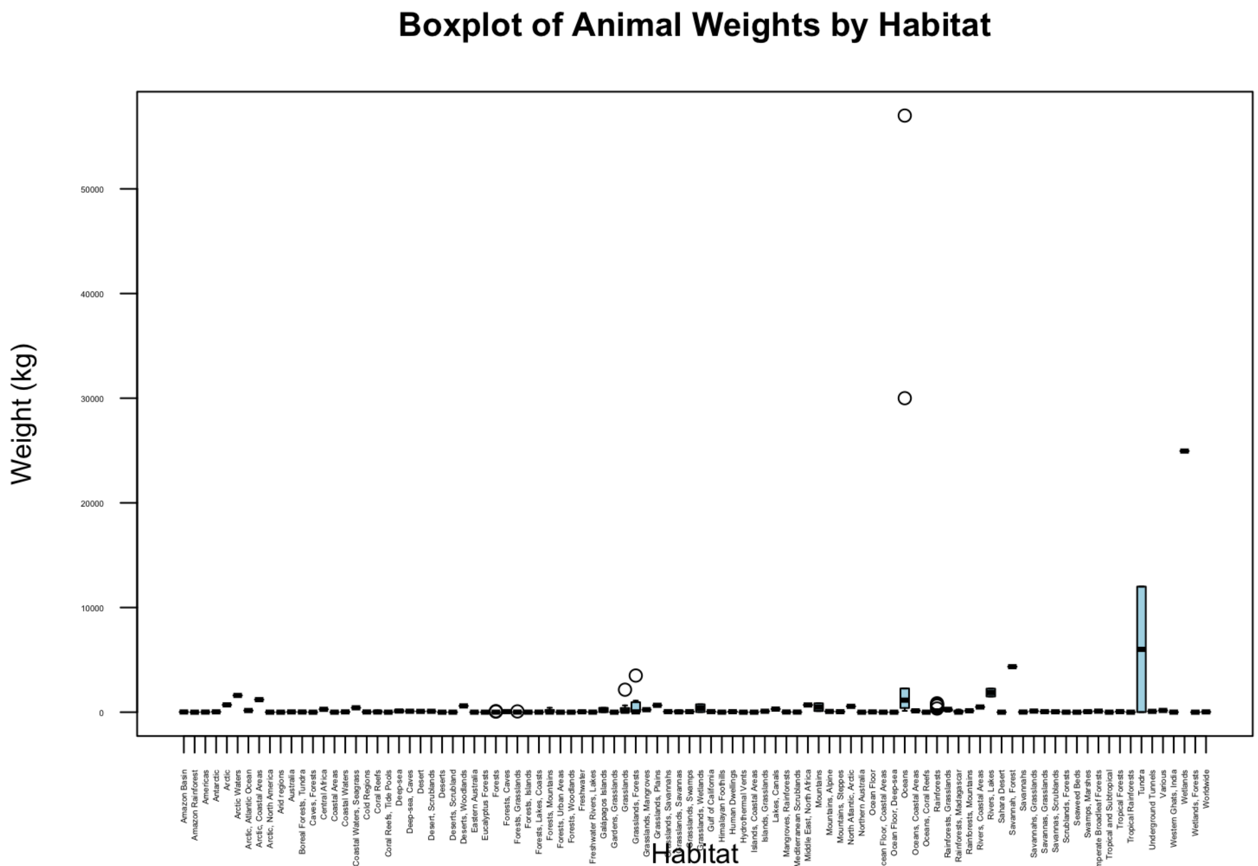
Height Distribution

- Most animals are clustered around moderate (50-100 cm) heights.



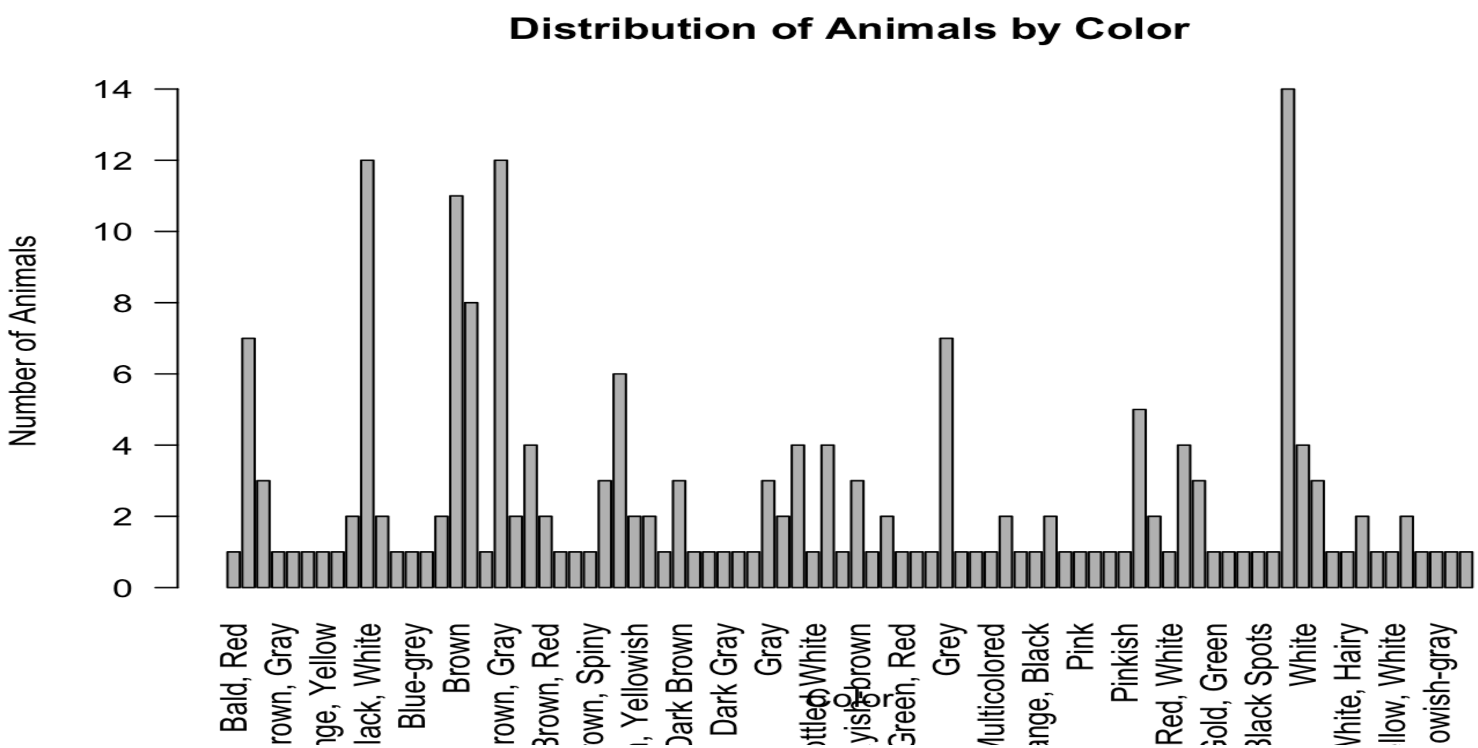
Habitats and Weights

We can see the Tundra habitat has the most weighted animals.



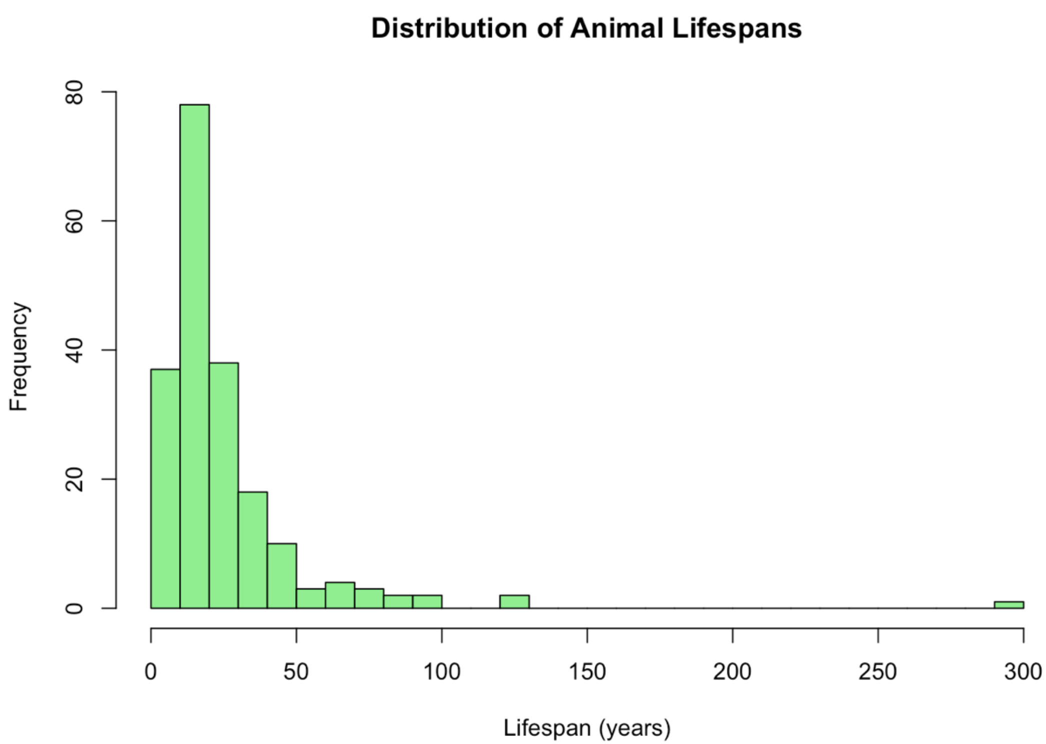
Color Distribution

- Black, White, Brown, Gray and their combinations are the most common colors.



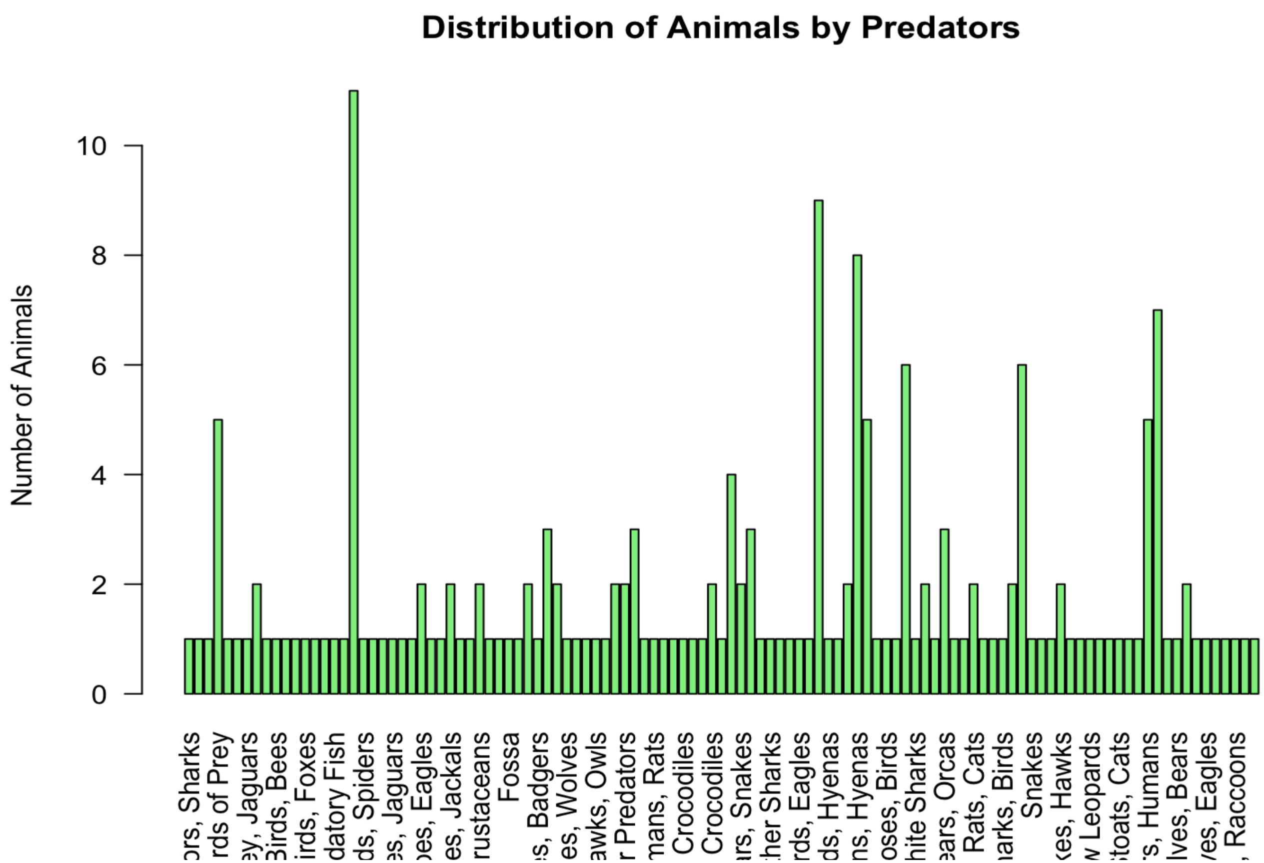
Lifespan Distribution

- Majority of animals live around 10-20 years. Exceptions exists.



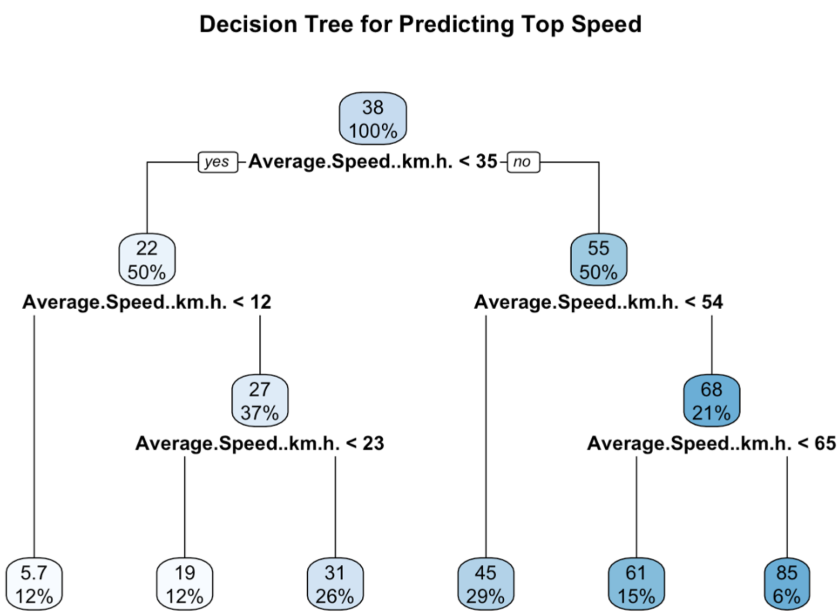
Predator Distribution

- Birds and snakes are the most frequent predators, followed by leopards and humans.



Predictive Modeling: Decision Tree

- Predicting 'Top Speed' based on numeric features.
- Model: Decision Tree.




```
> # Print Decision Tree performance
> cat("Decision Tree Model Performance:\n")
Decision Tree Model Performance:
> cat("RMSE:", round(tree_rmse, 2), "\n")
RMSE: 10.71
> cat("MAE:", round(tree_mae, 2), "\n")
MAE: 6.31
> cat("R-squared:", round(tree_r2, 2), "\n")
R-squared: 0.72
```

```
                          Overall
Average.Speed..km.h.    2.7169244
Gestation.Period..days. 0.5569686
Height..cm.             0.5688966
Lifespan..years.        0.4239857
Offspring.per.Birth     0.3939649
Weight..kg.             0.3671467
```

Conclusion

- - Cleaned messy data
- - Identified significant outliers
- - Visualized major patterns
- - Built and evaluated a predictive model

Note: The dataset is not useful to predicting. And also not good for visualizing the data because of so much unformatted data, I tried my best :).

Cleaning Codes

```r
animals_data <- read.csv('Animal Dataset.csv')
View(animals_data)
str(animals_data)

#remove the upto and give the exact value
cols_contains_upto <- c('Height..cm.', 'Weight..kg.', 'Lifespan..years.')

for(col in cols_contains_upto){
  match <- grepl("^Up to", animals_data[[col]])
  split_values <- strsplit(animals_data[[col]][match], " ")
  animals_data[[col]][match] <- sapply(split_values, function(parts) parts[3])
}
```

```r
#find the - values and take the mean of them
#lets consider height and there is only one entry which is entered in meter
col <- 'Height..cm.'
match <- grepl("-", animals_data[[col]])
split_values <- strsplit(animals_data[[col]][match], "-")
animals_data[[col]][match] <- sapply(split_values, function(x) (as.numeric(x[1]) + as.numeric(x[2])) / 2)

#BlueWhale which is entered as 33.5m(not in cm)
animals_data[33, col] <- 3350
```

Cleaning Codes

```r
# lets conisder weight as in kg and take the mean for no unit change
# there are some values which are represneted with comma
# so we are not taking the comma value (only two entries will be hardy entred with hand)
col <- 'Weight..kg.'
match <- grepl("-", animals_data[[col]]) & !grepl(",", animals_data[[col]])
split_values <- strsplit(animals_data[[col]][match], "-")
animals_data[[col]][match] <- sapply(split_values, function(x) (as.numeric(x[1]) + as.numeric(x[2])) / 2)

#handled the comma issue.
match <- grepl(",", animals_data[[col]])
animals_data[[col]][match]
which(match)
#fix
animals_data[8, col] <- 659
animals_data[28, col] <- 1100

#we have to conver the data to the numeric but there are other values.
```

strsplit(x, split, fixed = FALSE, perl = FALSE, useBytes = FALSE)

```r
#lets consider lifespan years
col<- 'Lifespan..years.'
match <- grepl("-", animals_data[[col]]) & !grepl("months|weeks|days", animals_data[[col]])
split_values <- strsplit(animals_data[[col]][match], "-")
animals_data[[col]][match] <- sapply(split_values, function(x) (as.numeric(x[1]) + as.numeric(x[2])) / 2)

#problem span
match <- grepl("months|weeks|days", animals_data[[col]])
which(match)
animals_data[[col]][match]
#fix
animals_data[32, col] <- 0.02
animals_data[35, col] <- 0.6
animals_data[77, col] <- 0.035
```

Cleaning Codes

```r
#lets consider avg speed and top speed values there are - values

cols <- c('Average.Speed..km.h.', 'Top.Speed..km.h.')
for(col in cols){
  match <- grepl("-", animals_data[[col]])
  split_values <- strsplit(animals_data[[col]][match], "-")
  animals_data[[col]][match] <- sapply(split_values, function(x) (as.numeric(x[1]) + as.numeric(x[2])) / 2)
}
```

```r
#lets consider 'Offspring.per.Birth' to manage the up to keyword
col <- 'Offspring.per.Birth'
match <- grepl("^Up to", animals_data[[col]])
split_values <- strsplit(animals_data[[col]][match], " ")
animals_data[[col]][match] <- sapply(split_values, function(parts) parts[3])
```

Cleaning Codes

```
#lets consider 'Gestation.Period..days.' to
col <- 'Gestation.Period..days.'

#only the valid but - values
match <- grepl("-", animals_data[[col]]) & !grepl("[A-Za-z]", animals_data[[col]])
split_values <- strsplit(animals_data[[col]][match], "-")
animals_data[[col]][match] <- sapply(split_values, function(x) (as.numeric(x[1]) + as.numeric(x[2])) / 2)

#months but with - values
match_months <- grepl("months", animals_data[[col]], ignore.case = TRUE)
split_values <- strsplit(animals_data[[col]][match_months], " ")
animals_data[[col]][match_months] <- sapply(split_values, function(x){
  months_parts <- strsplit(x[[1]], "-")[[1]]
  if (length(months_parts) == 2) {
    days <- (as.numeric(months_parts[1]) * 30 + as.numeric(months_parts[2]) * 30) / 2
  } else if (length(months_parts) == 1) {
    days <- as.numeric(months_parts[1]) * 30
  } else {
    days <- NA
  }
  return(days)
})
#weeks but with - values
match_months <- grepl("weeks|week", animals_data[[col]], ignore.case = TRUE)
split_values <- strsplit(animals_data[[col]][match_months], " ")
animals_data[[col]][match_months] <- sapply(split_values, function(x){
  months_parts <- strsplit(x[[1]], "-")[[1]]
  if (length(months_parts) == 2) {
    days <- (as.numeric(months_parts[1]) * 7 + as.numeric(months_parts[2]) * 7) / 2
  } else if (length(months_parts) == 1) {
    days <- as.numeric(months_parts[1]) * 7
  } else {
    days <- NA
  }
  return(days)
})
```

Cleaning Codes

```
#we also have million years value :d and with another explained value
animals_data[133, col] <- 10
animals_data[48, col] <- 120450000000000

#lets consider the 'Offspring.per.Birth' column
col <- 'Offspring.per.Birth'
# to handle normal - values
match <- grepl("-", animals_data[[col]]) & !grepl("[A-Za-z]", animals_data[[col]])
split_values <- strsplit(animals_data[[col]][match], "-")
animals_data[[col]][match] <- sapply(split_values, function(x) (as.numeric(x[1]) + as.numeric(x[2])) / 2)

match <- grepl("usually|approx.|rarely", animals_data[[col]], ignore.case = TRUE)
split_values <- strsplit(animals_data[[col]][match], " ")
animals_data[[col]][match] <- sapply(split_values, function(x){
  normal_parts <- strsplit(x[[1]], "-")[[1]]
  if (length(normal_parts) == 2) {
    part <- (as.numeric(normal_parts[1])  + as.numeric(normal_parts[2])) / 2
  } else if (length(normal_parts) == 1) {
    part <- as.numeric(normal_parts[1])
  } else {
    part <- NA
  }
  return(part)
})
```

Cleaning Codes

```r
# lets fill the varies and Not Applicable in all cols to NA
animals_data[animals_data == "Varies" | animals_data == "Not Applicable"] <- NA
# also fill the string values with a arbitrary number
animals_data[animals_data == "Thousands"] <- 5000
animals_data[animals_data == "Hundreds"] <- 500

match_comma <- grepl(",", animals_data[[col]])
which(match_comma)
animals_data[[col]][match_comma]

animals_data[40, col] <- 1000
animals_data[47, col] <- 1000
animals_data[74, col] <- 200000
animals_data[130, col] <- 10000
```

```r
# Convert all numeric columns to numeric type
numeric_cols <- c('Height..cm.', 'Weight..kg.', 'Lifespan..years.',
                  'Average.Speed..km.h.', 'Top.Speed..km.h.',
                  'Gestation.Period..days.', 'Offspring.per.Birth')

for(col in numeric_cols) {
  animals_data[[col]] <- as.numeric(animals_data[[col]])
}

# Save the cleaned dataset to a new CSV file
write.csv(animals_data, "Animal Dataset Cleaned.csv", row.names = FALSE)
```

Analyzing Codes

```r
# Import the dataset
animal_data <- read.csv("Animal Dataset Cleaned.csv")
original_data <- read.csv("Animal Dataset.csv")
View(animal_data)

# Identify the height outliers
height_outliers <- boxplot.stats(animal_data$Height..cm.)$out
outlier_animals_height <- animal_data[animal_data$Height..cm. %in% height_outliers, c("Animal", "Height..cm.")]
boxplot(animal_data$Height..cm., main = "Boxplot of Animal Heights", ylab = "Height (cm)")
mtext("Outliers are shown as points")
View(outlier_animals_height)

# Identify the weight outliers
weight_outliers <- boxplot.stats(animal_data$Weight..kg.)$out
outlier_animals_weight <- animal_data[animal_data$Weight..kg. %in% weight_outliers, c("Animal", "Weight..kg.")]
boxplot(animal_data$Weight..kg., main = "Boxplot of Animal Weights", ylab = "Weight (kg)")
mtext("Outliers are shown as points")
View(outlier_animals_weight)
```

```r
# Identify the weight and height outliers
weight_height_outliers <- animal_data[animal_data$Height..cm. %in% height_outliers & animal_data$Weight..kg. %in% weight_outliers, ]
plot(animal_data$Height..cm., animal_data$Weight..kg., main = "Weight vs Height",
     xlab = "Height (cm)", ylab = "Weight (kg)", col = "black", pch = 19)
points(weight_height_outliers$Height..cm., weight_height_outliers$Weight..kg., col = "red", pch = 19)
View(weight_height_outliers)

# Identify the lifespan outliers
lifespan_outliers <- boxplot.stats(animal_data$Lifespan..years.)$out
outlier_animals_lifespan <- animal_data[animal_data$Lifespan..years. %in% lifespan_outliers, c("Animal", "Lifespan..years.")]
boxplot(animal_data$Lifespan..years., main = "Boxplot of Animal Lifespan", ylab="Lifespan (years)")
mtext("Outliers are shown as points")
View(outlier_animals_lifespan)
```

Analyzing Codes

```
# Identify the gestation period outliers
gperiod_outliers <- boxplot.stats(animal_data$Gestation.Period..days.)$out
outlier_animals_gpreiod <- animal_data[animal_data$Gestation.Period..days. %in% gperiod_outliers, c("Animal", "Gestation.Period..days.
boxplot(animal_data$Gestation.Period..days., main = "Boxplot of Animal Gestation Period", ylab="Gestation Period (days)")
mtext("Outliers are shown as points")
View(outlier_animals_gpreiod)

# Identify the average speed outliers
aspeed_outliers <- boxplot.stats(animal_data$Average.Speed..km.h.)$out
outlier_animals_aspeed <- animal_data[animal_data$Average.Speed..km.h. %in% aspeed_outliers, c("Animal", "Average.Speed..km.h.")]
boxplot(animal_data$Average.Speed..km.h., main = "Boxplot of Animal Average Speed", ylab="Average Speed (km/h)")
mtext("Outliers are shown as points")
View(outlier_animals_aspeed)

# Identify the top speed outliers
topspeed_outliers <- boxplot.stats(animal_data$Top.Speed..km.h.)$out
outlier_animals_topspeed <- animal_data[animal_data$Top.Speed..km.h. %in% aspeed_outliers, c("Animal", "Top.Speed..km.h.")]
boxplot(animal_data$Top.Speed..km.h., main = "Boxplot of Animal Top Speed", ylab="Top Speed (km/h)")
mtext("Outliers are shown as points")
View(outlier_animals_topspeed)

# Identify the offspring per birth
osperbirth_outliers <- boxplot.stats(animal_data$Offspring.per.Birth)$out
outlier_animals_osperbirth <- animal_data[animal_data$Offspring.per.Birth %in% Offspring.per.Birth, c("Animal", "Offspring.per.Birth")
boxplot(animal_data$Offspring.per.Birth, main = "Boxplot of Offspring Per Birth", ylab="Offspring per Birth")
mtext("Outliers are shown as points")
View(outlier_animals_osperbirth)
```

Analyzing Codes

```
# Height distribution
hist(animal_data$Height..cm., main = "Distribution of Animal Heights",xlab = "Height (cm)",ylab = "Frequency",col = "blue",border

# Weight distribution
hist(animal_data$Weight..kg., main = "Distribution of Animal Weights",xlab = "Weight (kg)",ylab = "Frequency",col = "green",border

# Lifespan distribution
hist(animal_data$Lifespan..years., main = "Distribution of Animal Lifespans",xlab = "Lifespan (years)",ylab = "Frequency",col = "l

# Diet and weight
hist(animal_data$Weight..kg., animals_data$Diet,main = "Scatter Plot of Height vs Weight with Height Outliers",xlab = "Height (cm)

# Color distribution
color_distribution <- table(animals_data$Color)
barplot(color_distribution,
        main = "Distribution of Animals by Color",
        xlab = "Color",
        ylab = "Number of Animals",
        las = 2)
```

```
# Predator distribution
predator_distribution <- table(animals_data$Predators)
barplot(predator_distribution,
        main = "Distribution of Animals by Predators",
        ylab = "Number of Animals",
        col = "lightgreen",
        las = 2)
View(predator_distribution)
```

Modeling Codes

```
# read data
model_data <- read.csv("Animal Dataset Cleaned.csv")
model_data <- model_data[sapply(model_data, is.numeric)]
model_data <- na.omit(model_data)

# Target var will be topspeed xd
target_variable <- "Top.Speed..km.h."

#train test split
set.seed(123)
train_index <- createDataPartition(model_data[[target_variable]], p = 0.8, list = FALSE)
train_data <- model_data[train_index, ]
test_data <- model_data[-train_index, ]
```

```
# Train a Decision Tree model
tree_model <- rpart(Top.Speed..km.h. ~ ., data = train_data)

# Plot the decision tree
rpart.plot(tree_model, main = "Decision Tree for Predicting Top Speed")

# Make predictions on the test set
tree_predictions <- predict(tree_model, newdata = test_data)

# Evaluate Decision Tree model
tree_rmse <- rmse(test_data$Top.Speed..km.h., tree_predictions)
tree_mae <- mae(test_data$Top.Speed..km.h., tree_predictions)
tree_r2 <- R2(tree_predictions, test_data$Top.Speed..km.h.)

# Print Decision Tree performance
cat("Decision Tree Model Performance:\n")
cat("RMSE:", round(tree_rmse, 2), "\n")
cat("MAE:", round(tree_mae, 2), "\n")
cat("R-squared:", round(tree_r2, 2), "\n")
```

Thank you!