Visualizing Variable Effects in Black Box Models

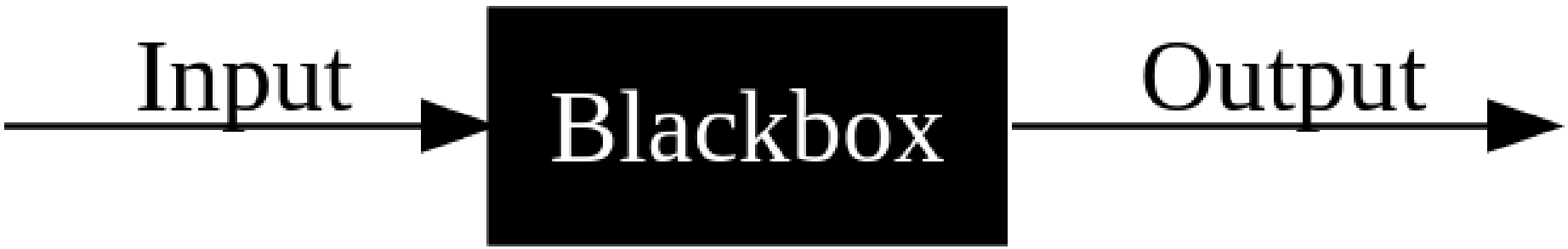
Saint Louis R User Group

John Snyder

December 4th, 2019

Black Boxes

- Some machine learning models are called "black boxes" for a reason.
  - Difficult to interpret parameters
- Neural networks, Random forests, SVMs are wildly popular
  - High predictive accuracy.
  - Non-parametric.
- These techniques acheive their predictive superiority through their nonparametric + nonlinear nature.
- This is the root of the interpretability vs flexability tradeoff.



Reclaiming Interpretability

Various different model-agnostic methods exist to bring back the interpretive aspect of modeling.

- Variable importance
- Local interpretable model-agnostic explanations (LIME)
  - STL-RUG January 2019
- Plotting Feature effects - *Today*
  - Partial Dependence Plot (PDP)
  - Individual Conditional Expectation (ICE)
  - Accumulated Local Effects (ALE)
- The iml package provides an interface to all of these.

```
install.packages("iml")  # CRAN
devtools::install_github("christophM/iml") # Development
library(iml)
```

Meet our Dataset

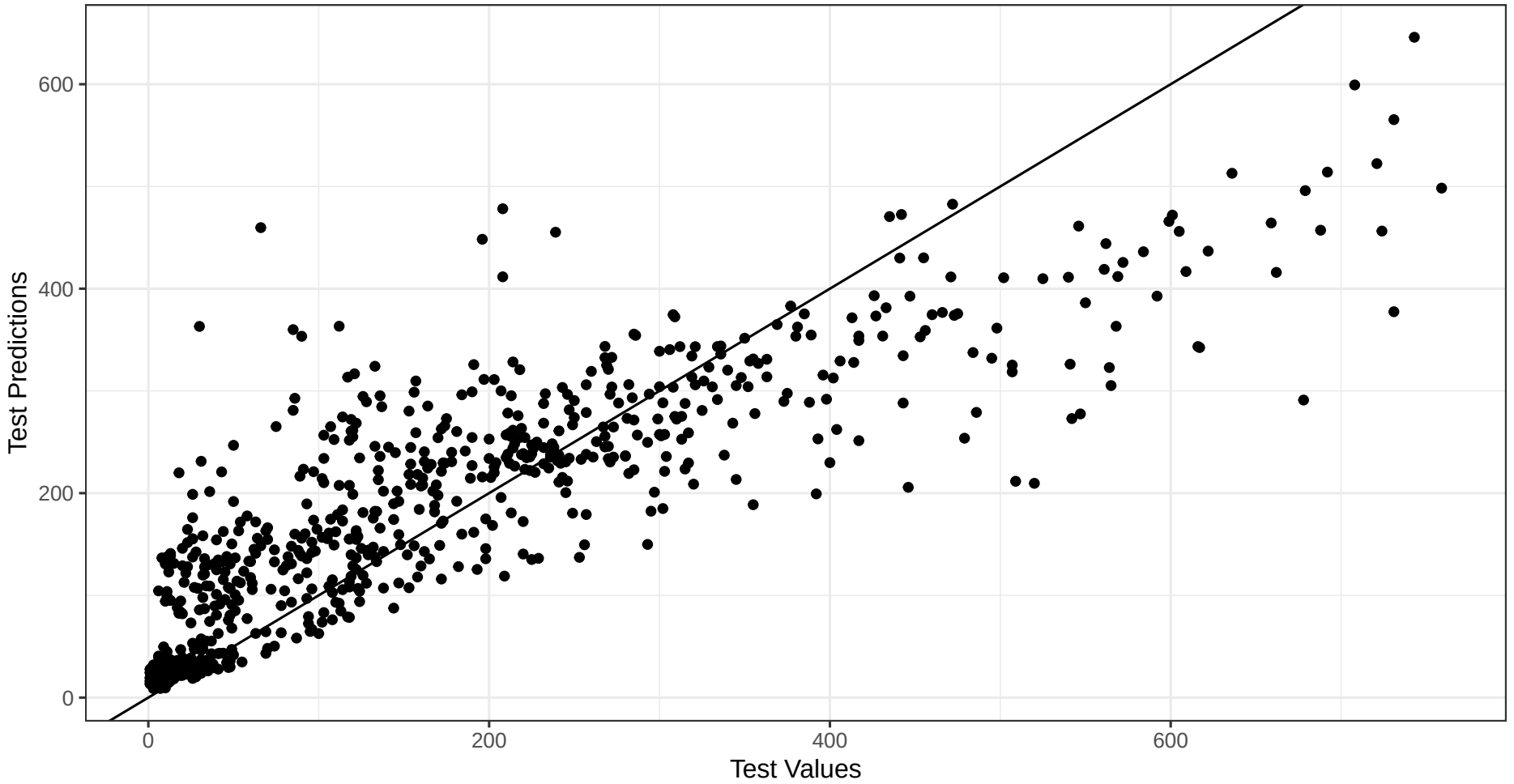
Bikesharing rental counts from Washington DC

- archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset
- Hourly data, bike rentals and weather information

```
Bike <- read.csv("../hour.csv") %>%
  select(cnt,                  # Number of Rentals, the Response
         yr,                   # Year (0: 2011, 1: 2012)
         mnth,                 # Month (1 to 12)
         hr,                   # Hour (0 to 23)
         holiday,              # Whether day is holiday or not
         weekday,              # Day of the week, 0 = Sunday
         workingday,           # 1 if neither weekend nor holiday.
         weathersit,           # Ordinal weather indicator
         temp,                 # Standardized Temperature
         hum,                  # Humidity divided to 100
         atemp,                # Standardized Apparent Temperature
         windspeed)            # Normalized wind speed
```

Build a Super Cool ML Predictor

```
Bike.Test  <- Bike %>% filter(Year==1 & Month == 12)
Bike.Train <- Bike %>% filter(!(Year==1 & Month == 12))
rf <- randomForest(Count ~ ., data = Bike.Train, ntree = 500)
```



Build iml Predictor

```
rf <- randomForest(Count ~ ., data = Bike, ntree = 500)
X <- Bike %>% select(-Count)
Y <- Bike %>% select(Count)
Pred_Obj <- Predictor$new(rf, data = X, y = Y)
```

We initialize a new predictor by passing:

- Model: Any model with with an S3 predict function.
  - High compatability with mlr and caret objects.
  - Keras works out of the box
- X and Y data.
- Specific arguments if needed for flexability.
  - Needed for nonstandard modeling software
  - Custom predict.fun
  - prediction type

NOTE: The iml package uses R6 objects, so syntax may look odd

First Step: Feature Importance

The model error is measured before and after shuffling the values of the feature.

- Shuffling should be repeated for accuracy.
- Parallel backend is builtin.

```
library(doParallel)
cl = makeForkCluster(6)
registerDoParallel(cl)
```

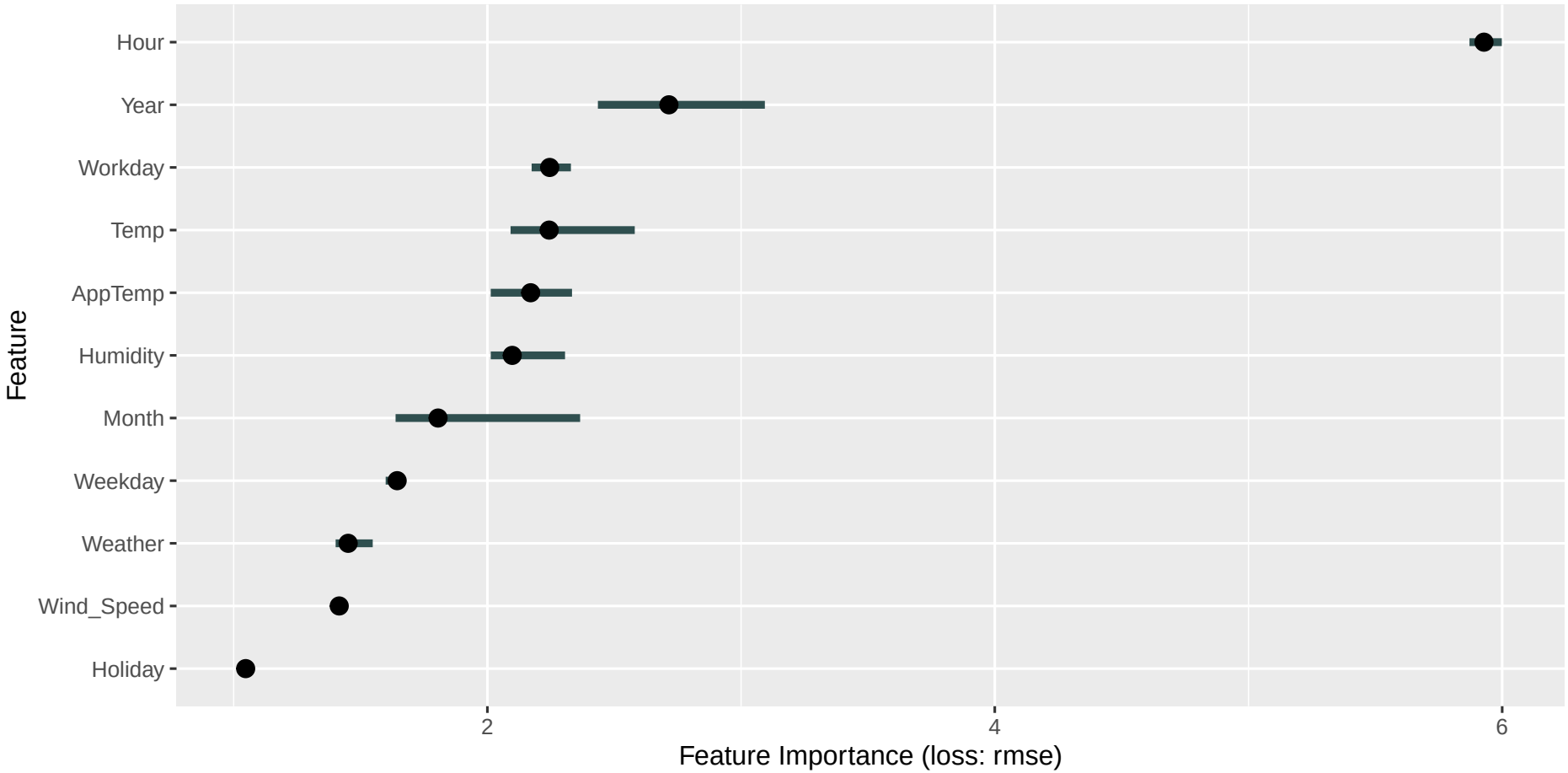
We can then create the feature importance object.

```
VarImp <- FeatureImp$new(Pred_Obj,
                         loss = "rmse",
                         parallel = TRUE,
                         n.repetitions = 30)
```

Any guesses on what will be important?

Variable Importance Plot [1]

```
VarImp$plot()
```



This does not tell us about the *nature* of these relationships.

[1] The point is *median* importance over the repetitions.

Partial Dependence Plots (PDP)

- Visualise the effect of variables on predicted values.
- Friedman 2001
  - Popular, commonly integrated

Idea:

1. Pick one or two variables.
2. Average out the effect of all other variables for a given value of our selected value.

Advantages:

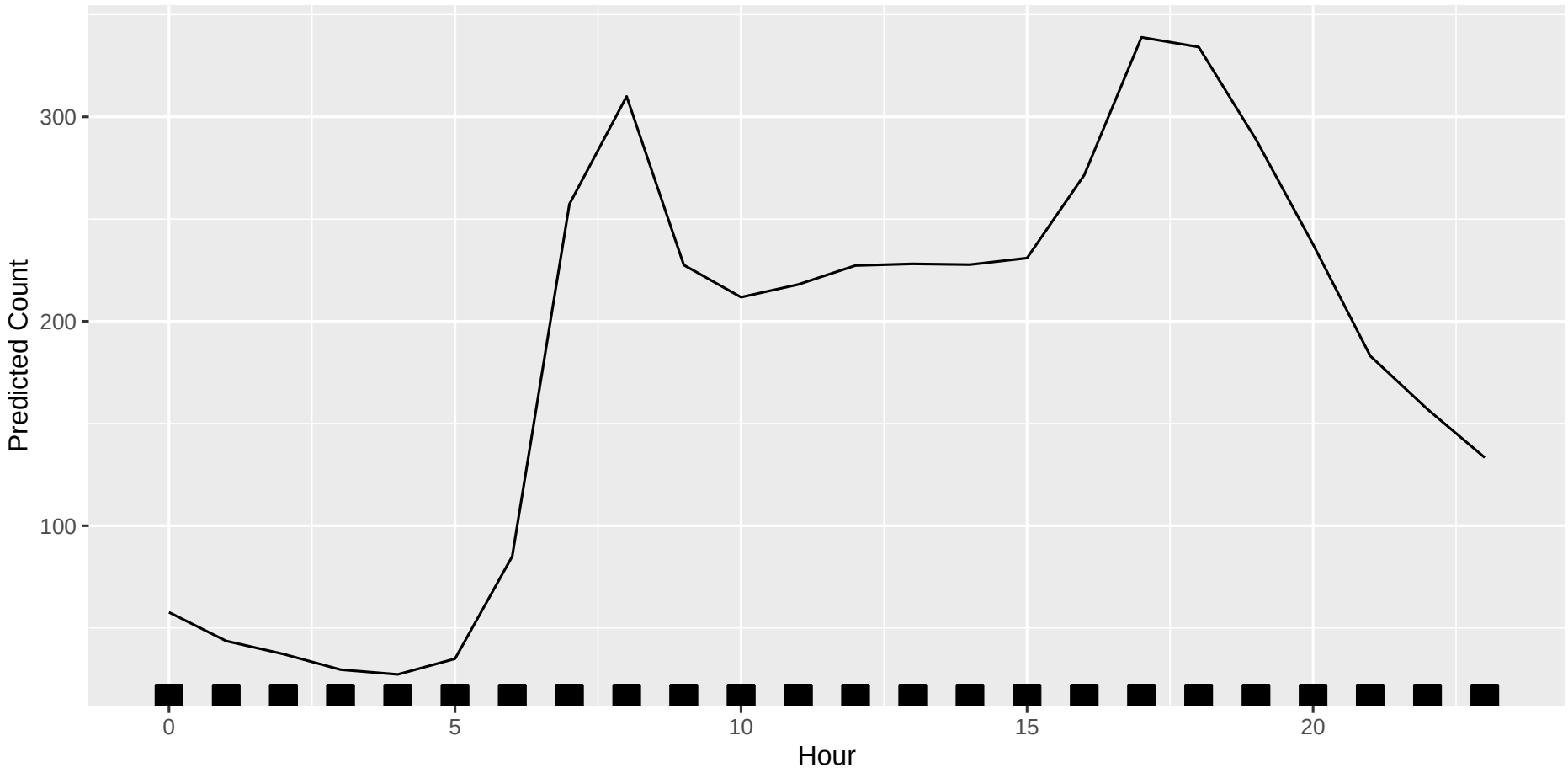
- Easy to understand the computation.
- Works *great* if variables are uncorrelated!

Disadvantages:

- Variables are usually correlated.
- Effects can be masked by interaction.

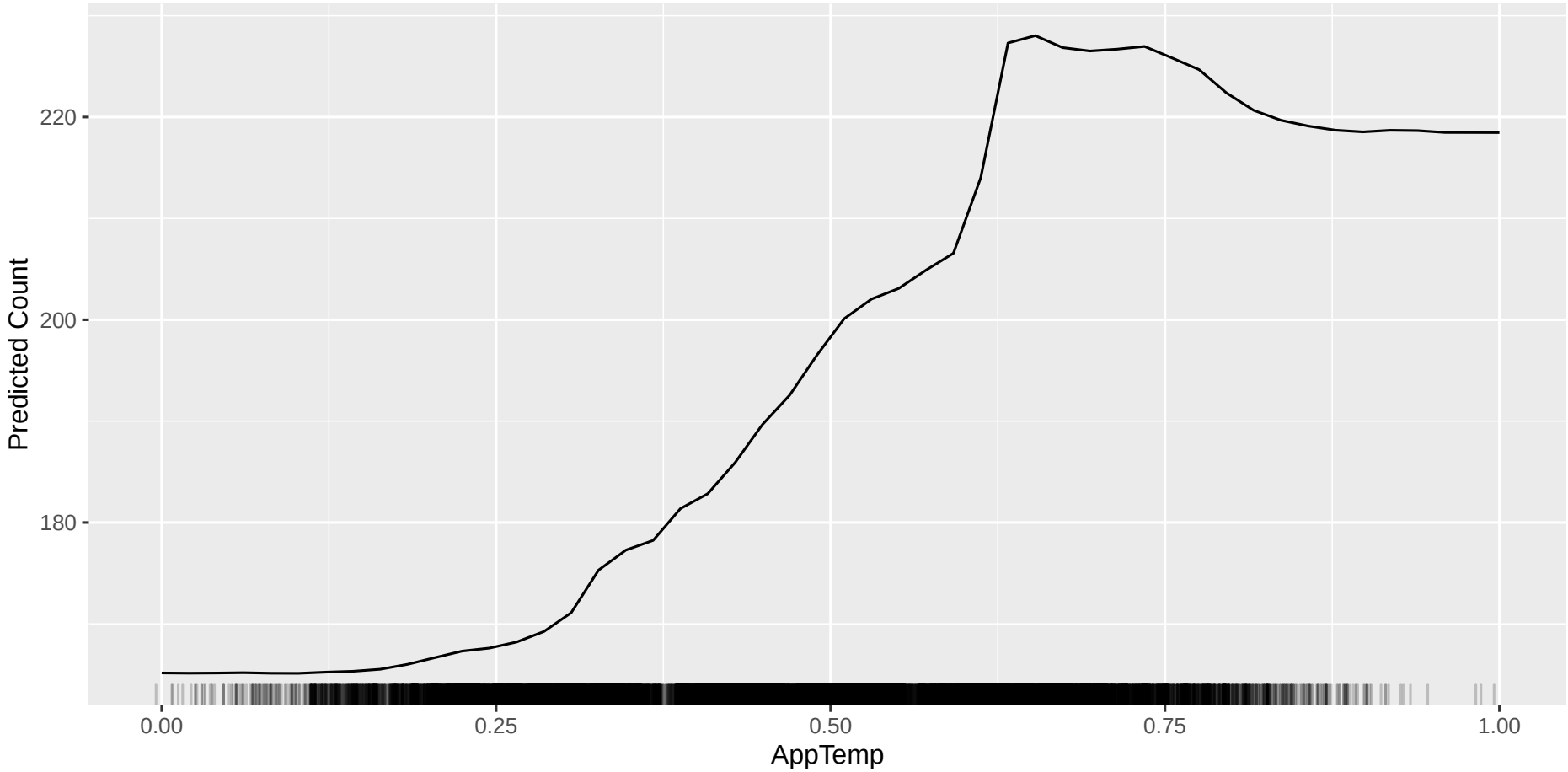
PDP - Time of Day

```
PDP <- FeatureEffect$new(Pred_Obj,feature = "Hour",
                         method = "pdp",grid.size = 24)
PDP$plot()
```



PDP - Apparent Temperature

```
PDP <- FeatureEffect$new(Pred_Obj,feature = "AppTemp",
                         method = "pdp",grid.size = 50)
PDP$plot()
```



Individual Conditional Expectation (ICE)

- Disaggregate PDPs - A line for each obseration!
  - Useful for finding heterogeneity in relationships
  - Usually caused by interactions between variables
- Goldstein et al. 2015

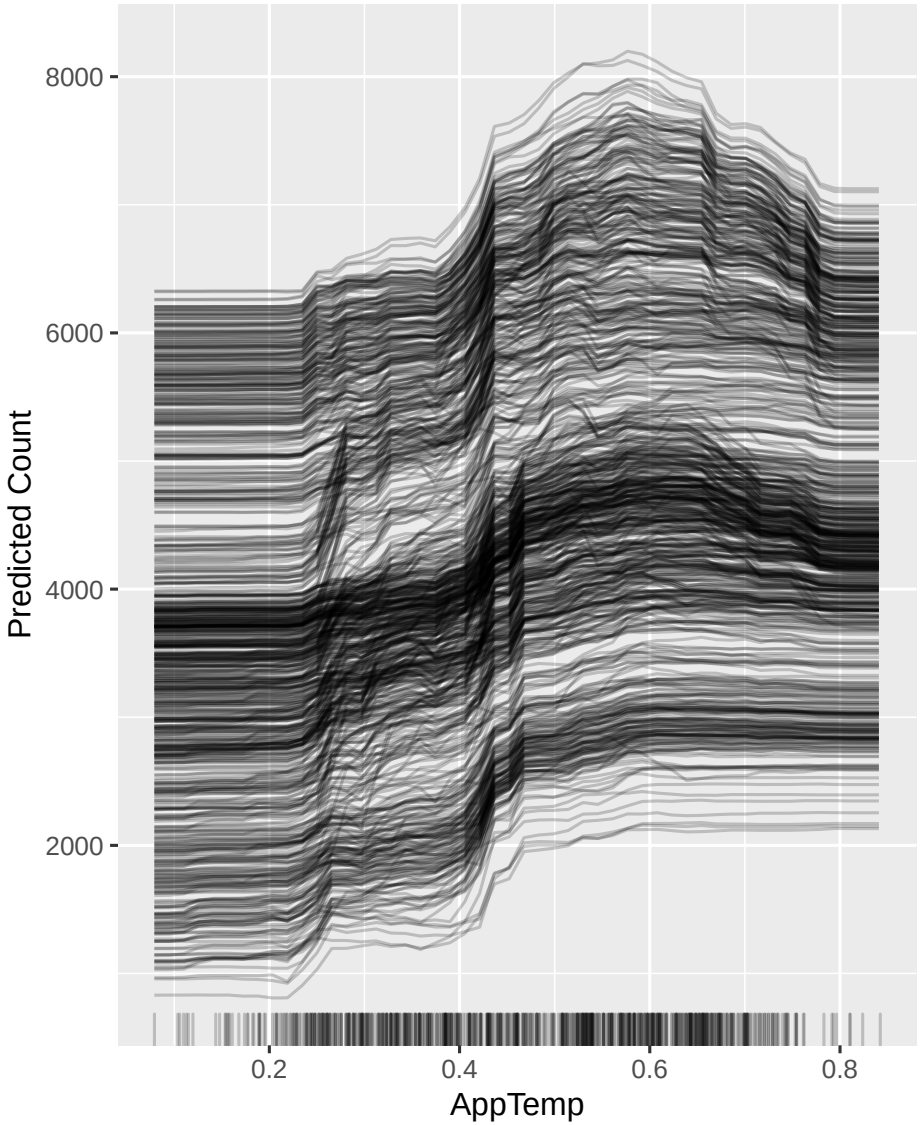
Advantages(over PDP):

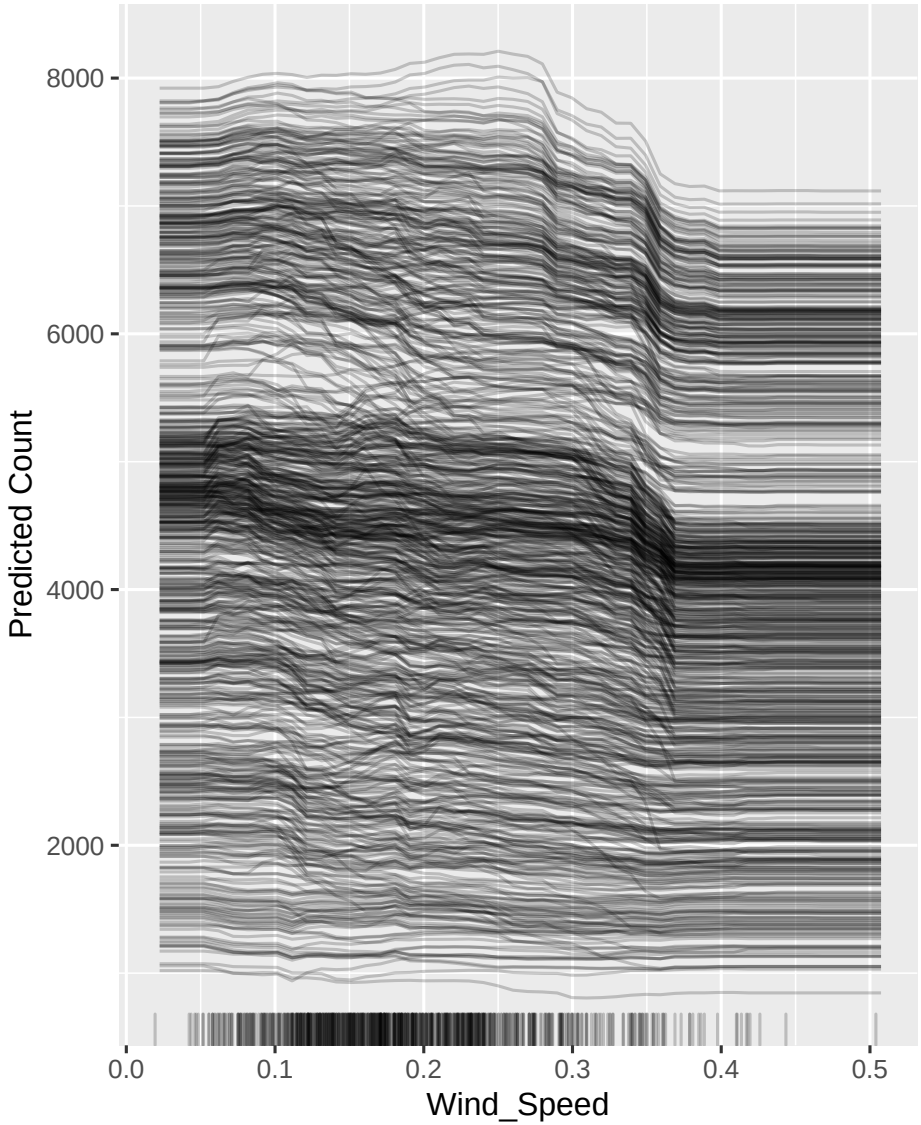
- Helps find potential interactions.

Disadvantages:

- Correlation problem again.
- For a large number of observations, usually the plot is a blob of lines

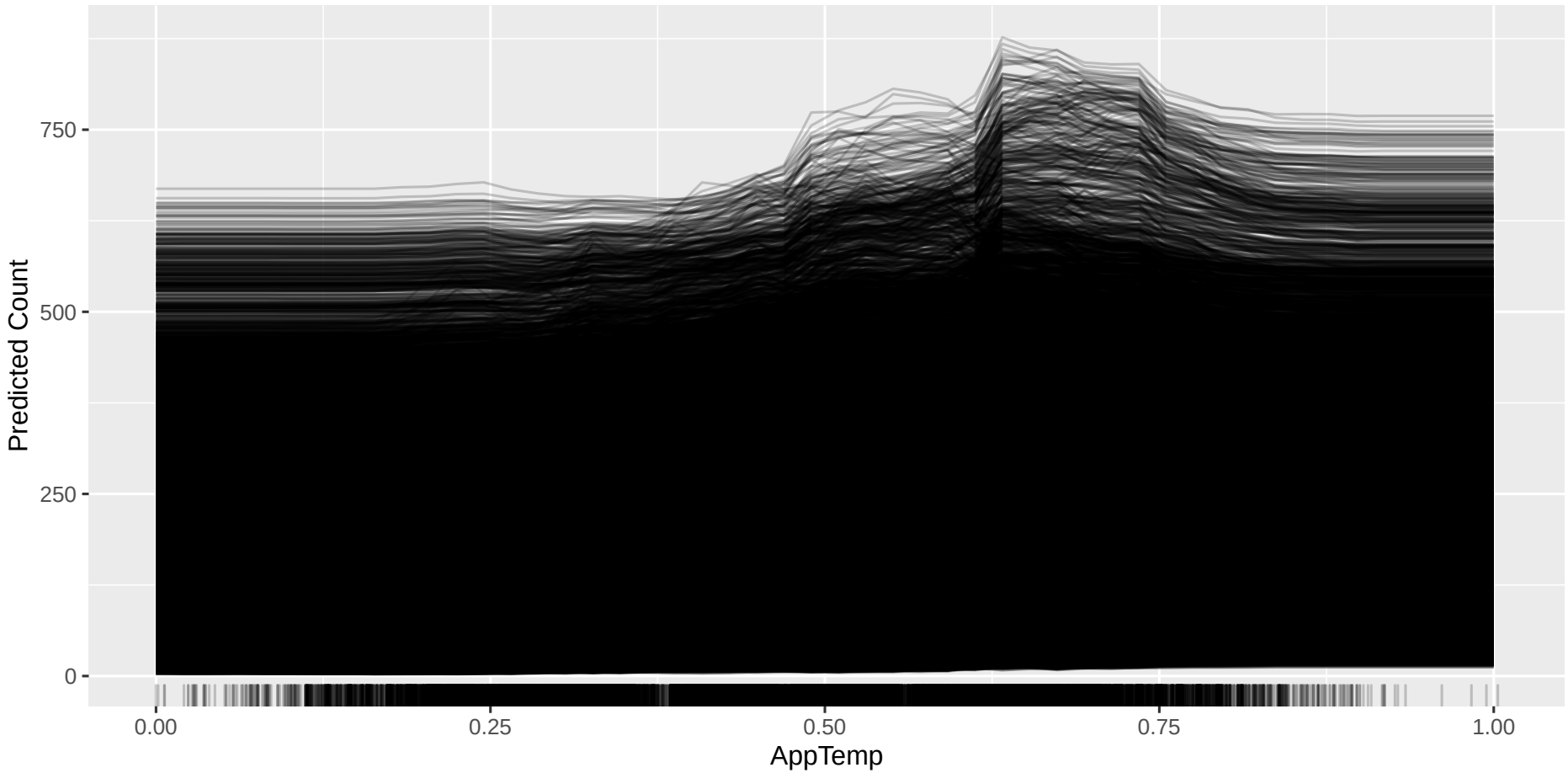
ICE Plot: Daily Dataset





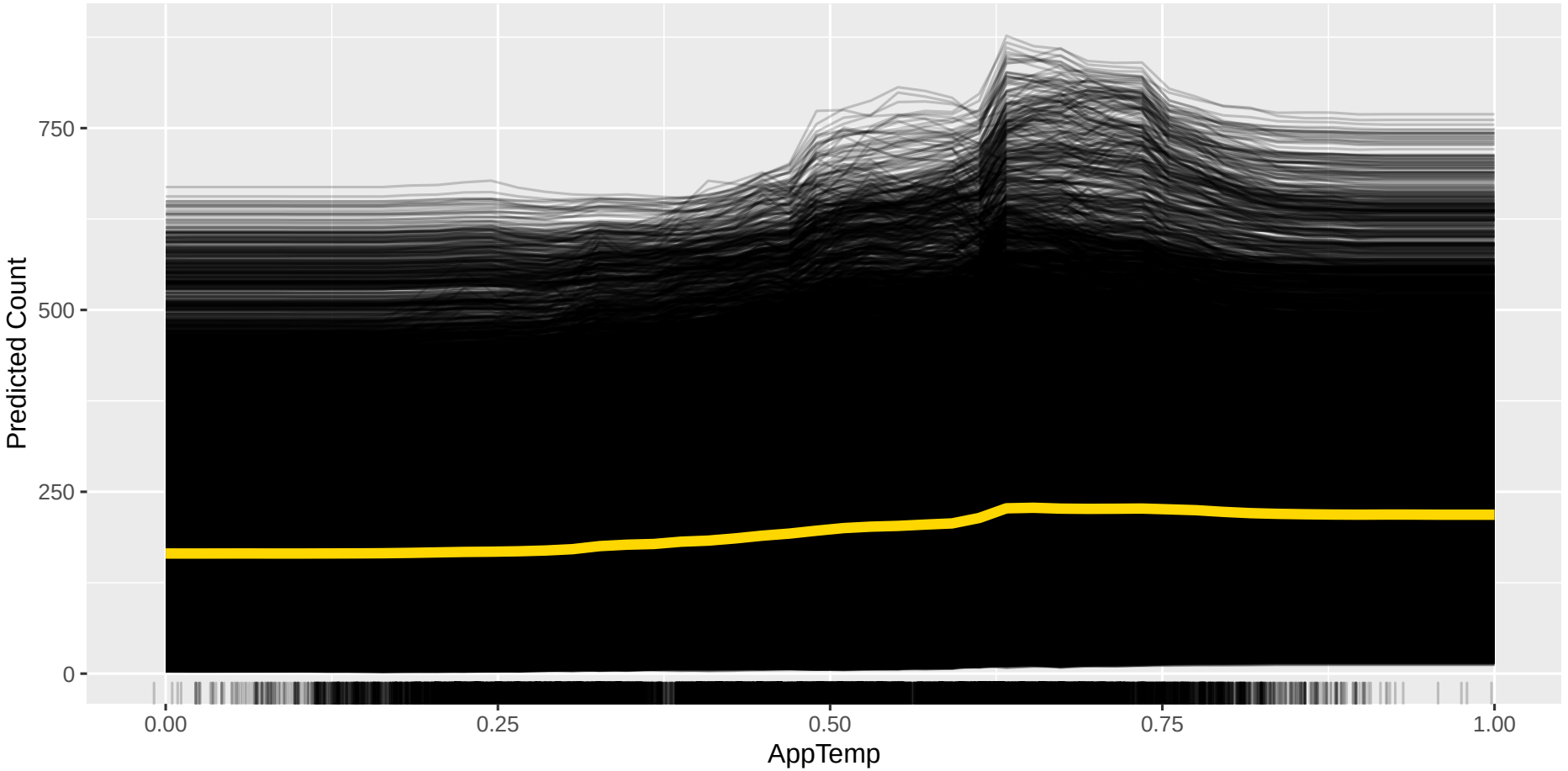
ICE Plot: Useless For Hourly Data

```
ICE <- FeatureEffect$new(Pred_Obj,feature = "AppTemp",
                         method = "ice",grid.size = 50)
ICE$plot()
```



ICE Plot with PDP overlay

```
ICE_plus_PDP <- FeatureEffect$new(Pred_Obj,feature = "AppTemp"
                                  method =  "pdp+ice",grid.size = 50)
ICE_plus_PDP$plot()
```



Interesting Take: ICE Plot Aggregation

```
ICE_Results <- ICE$results  #Extract values
head(ICE_Results)
```

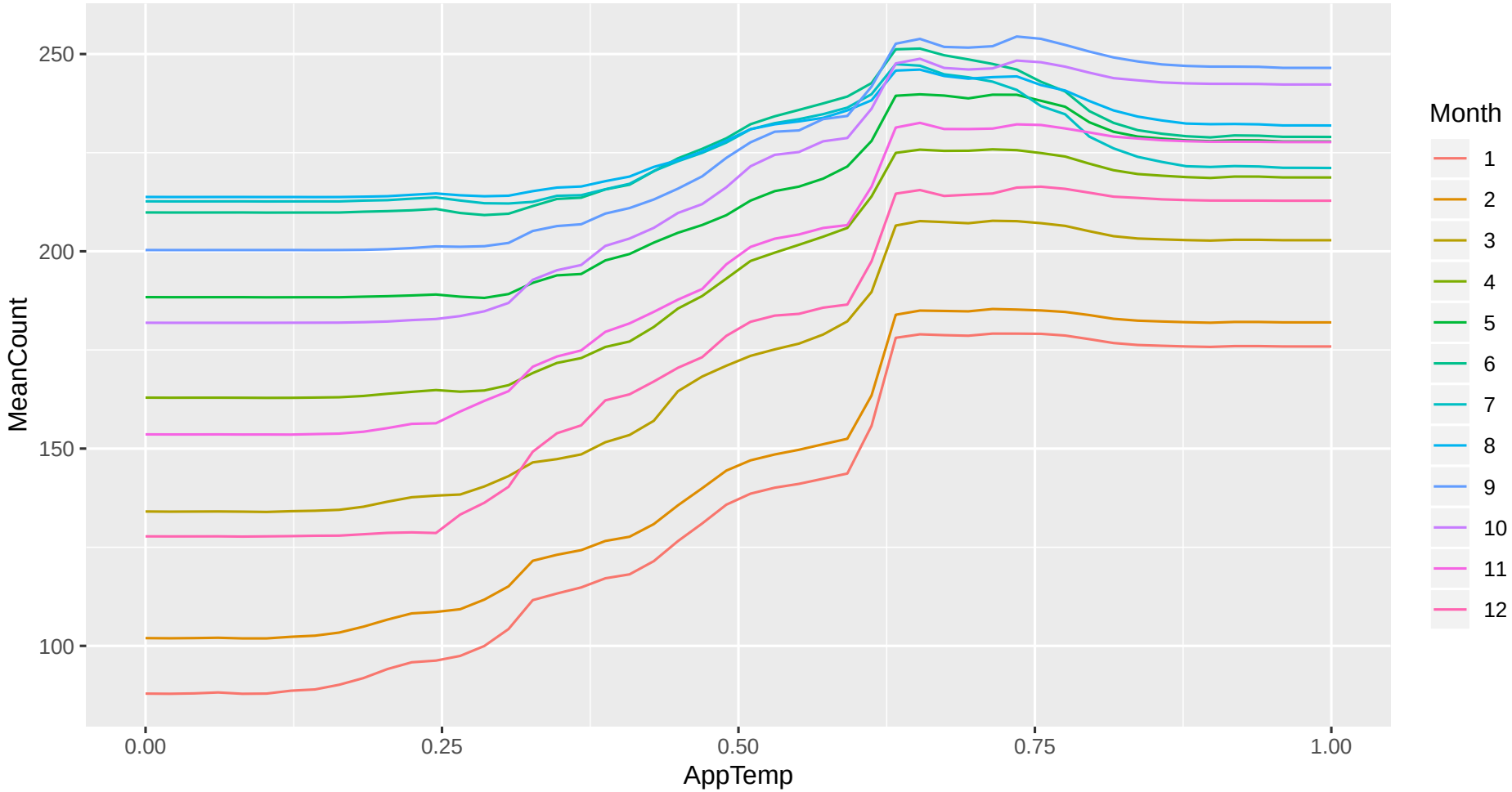
```
##       AppTemp   .y.hat .type .id
## 1 0.00000000 24.04901   ice   1
## 2 0.02040816 24.04001   ice   1
## 3 0.04081633 24.04001   ice   1
## 4 0.06122449 24.04901   ice   1
## 5 0.08163265 24.12494   ice   1
## 6 0.10204082 24.37969   ice   1
```

```
ICE_Results$Month <- as.factor(Bike$Month[ICE_Results$.id])

ICE_Agg <- ICE_Results %>%
  group_by(Month,AppTemp) %>%
  summarise(MeanCount = mean(.y.hat))
```

Interesting take: ICE plot 2

```
ggplot(ICE_Agg,aes(x=AppTemp,y=MeanCount,color=Month)) +
  geom_line()
```



Accumulated Local Effects (ALE)

ALE plots are a recent development focused on providing a new perspective.

- Built to overcome challenges with correlated variables
- Apley 2019

Divides the space into intervals and looks at differences in *predictions* within those buckets.

Advantages:

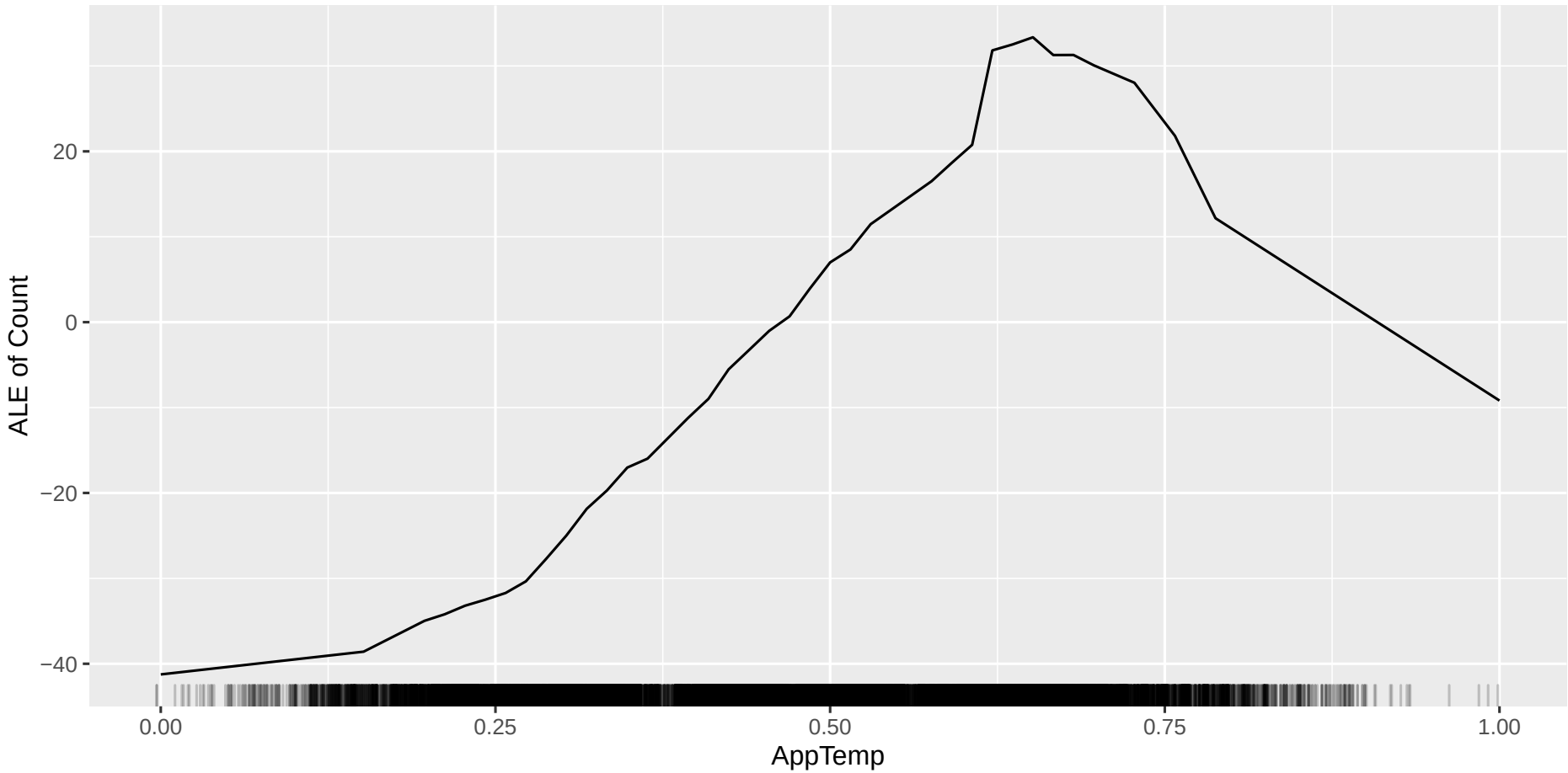
- Fast
- Easy to Interpret

Disadvantages:

- Tradeoff with the number of buckets
  - Too many intervals and the line is "jumpy"
  - Too few and the relationship is hidden

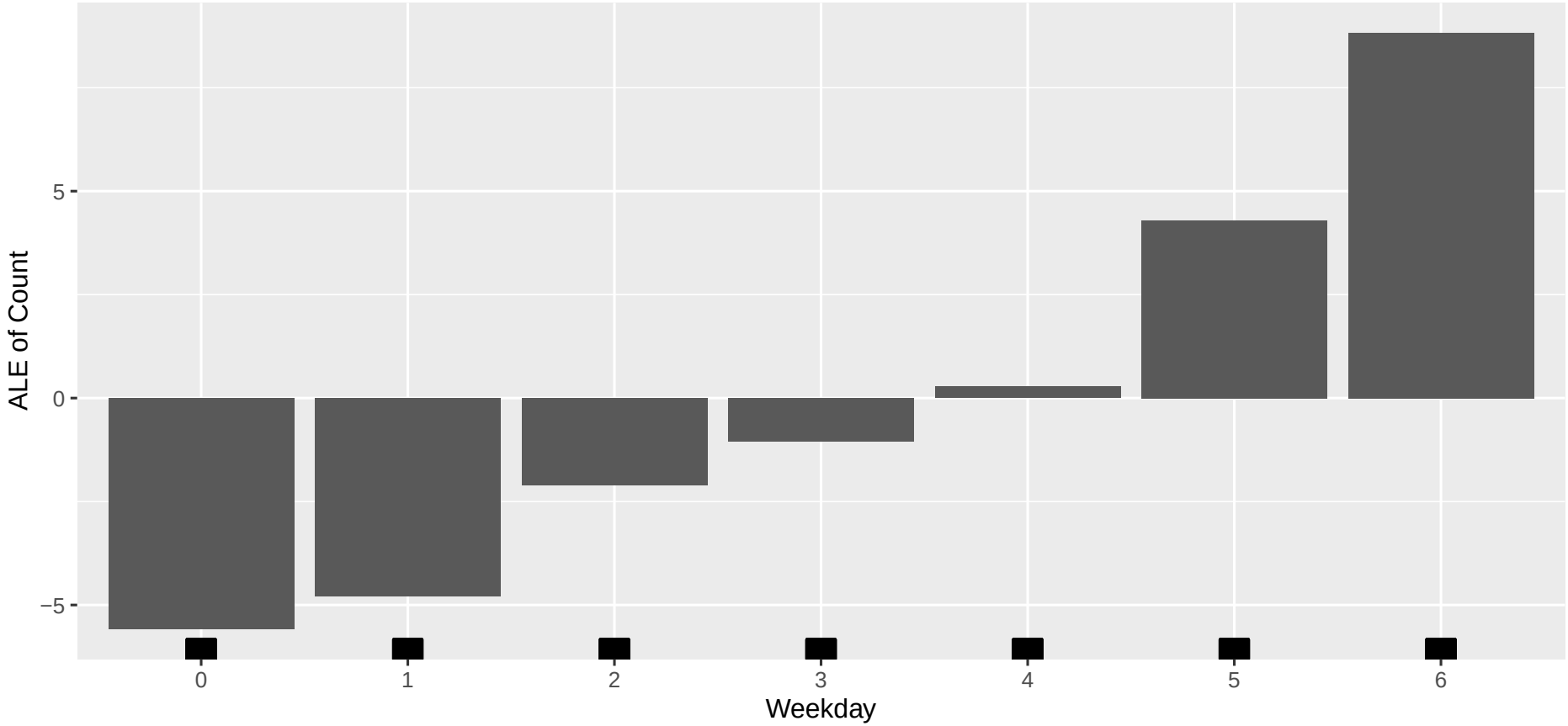
ALE Plot: Slightly Different than PDP!

```
ALE <- FeatureEffect$new(Pred_Obj,feature = "AppTemp",
                         method = "ale",grid.size = 50)
ALE$plot()
```



Categorical Predictors [2]

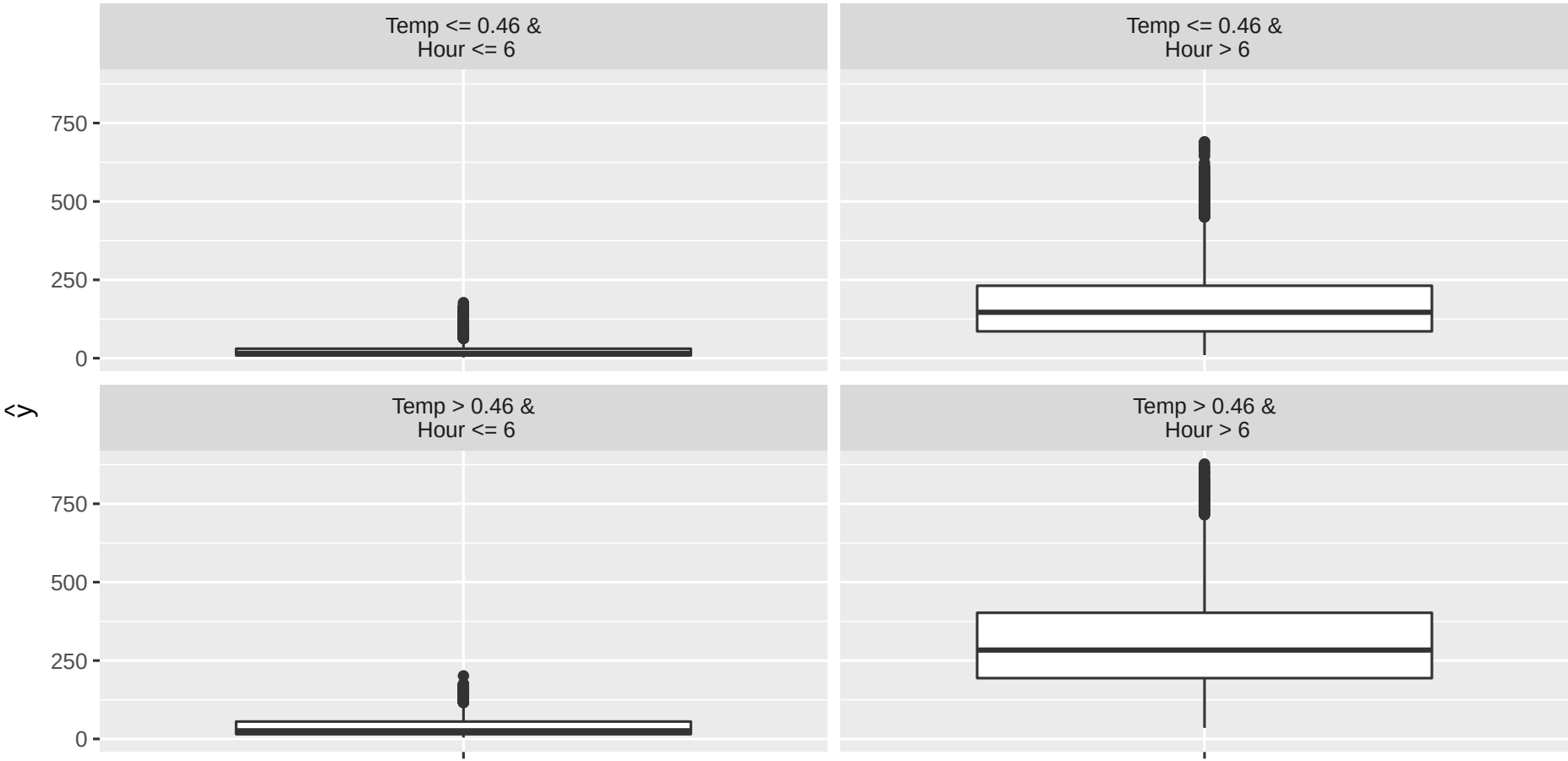
```
CatPred <- FeatureEffect$new(Pred_Obj,feature = "Weekday")
CatPred$plot()
```



[2] Post presentation note: Variable changed to ordinal after discussion. If a variable is a nominal factor, when the plot is requested the package will order the levels of the factor so that adjacent levels are similar with respect to all other variables. This doesn't make sense for ordinal variables like this. New plot looks awesome! 0 is Sunday.

Surrogate Tree Visualization

```
SurTree = TreeSurrogate$new(Pred_Obj, maxdepth = 2)
plot(SurTree)
```



Second Order ALE Plot

We can use the ALEPlot package directly to make a contour plot

- Assess *joint* relationships between variables

```
library(ALEPlot)
yhat <- function(X.model, newdata){
  as.numeric(predict(X.model, newdata))}
X <- Bike %>% select(-Count)
```

The package draws plots with a numeric argument "J" based on the name indecies

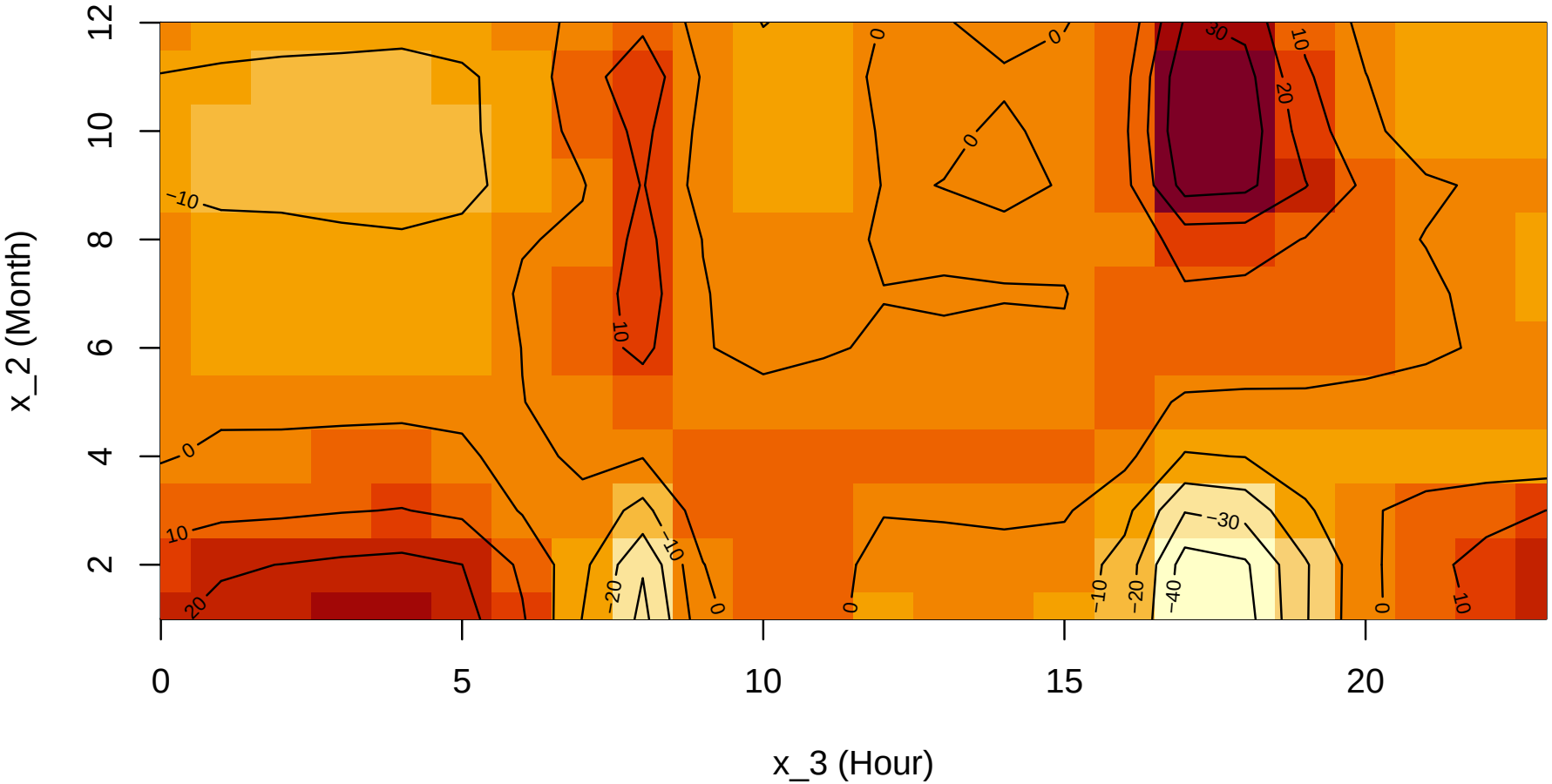
```
names(X)
```

```
##  [1] "Year"        "Month"       "Hour"        "Holiday"     "Weekday"
##  [6] "Workday"     "Weather"     "Temp"        "Humidity"    "AppTemp"
## [11] "Wind_Speed"
```

- J = c(2,3) will plot Month and Hour

Second Order ALE Plot: Time vs Month

```
ALEPlot(X, rf,pred.fun=yhat,J=c(3,2), K=50,NA.plot = TRUE)
```



John's Final Thoughts

- Visualization is an extremly powerful tool
  - Often allows us to verify intuition.
  - Explain results to stakeholders.
    - ★ a priori hypothesis confirmation.

- iml has a suite of tools for interpretation
  - Shapley: Shapley$new() function
  - LIME: LocalModel$new() function
  - Interaction investigator: Interaction$new()
  - Effect plot matrix: FeatureEffects$new()

- Still no "significance" measures or easy way to measure variation.

- Some packages need custom predict functions to work with iml