# Data Set Profile: Bike Sharing Demand

*Daniel Dittenhafer*

*Saturday, September 27, 2014*

| Data Set Characteristics | Number of Observations | Area | Attribute Characteristics | Number of Attributes | Missing Values? |
|---|---|---|---|---|---|
| Multivariate | 10,886 | Business | Categorical, Integer, Date/time, Decimal | 12 | No |

### Source

Kaggle.com Competition: Bike Sharing Demand: https://www.kaggle.com/c/bike-sharing-demand

The goal of the competition, from the Kaggle website, is to "predict the total count of bikes rented during each hour covered by the test set, using only information available prior to the rental period." The data set profiled herein is the training data set.

### Attribute Information

| Field | Data Type | Description |
|---|---|---|
| datetime | date & hour | First 19 days of each month Min: 01/01/2011 00:00; Max: 12/19/2012 23:00 |
| season | integer | Categorical; 1 = spring; 2 = summer; 3 = fall; 4 = winter; |
| holiday | boolean | 1 = a holiday; 0 = not a holiday; |
| workingday | boolean | 1 = a work day; 0 = weekend or holiday; |
| weather | integer | Categorical; 1) Clear, Few clouds, Partly cloudy, Partly cloudy; 2) Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist; 3) Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds; 4) Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog; |
| temp | decimal | temperature in degrees Celsius |
| atemp | decimal | apparent temperature in degrees Celsius |
| humidity | integer | relative humidity percentage |
| windspeed | decimal | the speed that air is moving in unknown units |
| casual | integer | the number of non-registered bike shares for the hour |
| registered | integer | the number of registered bike shares for the hour |
| count | integer | the total number of bike shares for the hour |

### Comments

There were no inherent character columns and the data, where appropriate, was already converted to factor-like integer values. As such, in order to better map the data set to this exercise, I have added a seasonName column and reverted the season column into this new column to begin with. Given the goal of the Kaggle competition to predict future bike share count, shifting the count values by a fixed period aids in this analysis. Additionally, for analysis purposes the datetime field would be better broken up into components including a simple integer hour of day, individual month value, day of week, segment of the day, etc. Some of these transformations are applied in the code segment that follows including the inclusion of a nextHourCount attribute which reflects the following hour's total bike rentals for a given hour.

```
# Load the data into a data.frame
csv_file <- file.path(projRoot, "Week5", "BikeSharingDemand.csv")
csv <- read.table(csv_file, header=TRUE, sep=",")

# Revert season to character data.
bikes <- data.frame(csv,
                    seasonName=NA, hourOfDay=NA,
                    dayOfWeek=NA, dayOfWeekInt=NA,
                    monthOfYear=NA, segmentOfDay=NA,
                    nextHourDateTime=NA, nextHourCount=NA)
bikes[bikes$season == 1,]$seasonName <- "spring"
bikes[bikes$season == 2,]$seasonName <- "summer"
bikes[bikes$season == 3,]$seasonName <- "fall"
bikes[bikes$season == 4,]$seasonName <- "winter"

# 14 Add an integer column for hour of day
bikes$hourOfDay <- lubridate::hour(bikes$datetime)

# 15 Add a factor and integer column for day of week
bikes$dayOfWeek <- as.factor(weekdays(strptime(as.character(bikes$datetime),
                                               format="%Y-%m-%d %H:%M:%S")))
# 16
bikes$dayOfWeekInt <- as.numeric(bikes$dayOfWeek)

# 17 Add an integer column for month of year
bikes$monthOfYear <- lubridate::month(bikes$datetime)

# 18 Add an integer column for segment of day
bikes$segmentOfDay <- ifelse(bikes$hourOfDay >= 5 & bikes$hourOfDay < 12,
                             1, # Morning
                             ifelse(bikes$hourOfDay >= 12 & bikes$hourOfDay < 17,
                                    2, # Afternoon
                                    ifelse(bikes$hourOfDay >= 17 & bikes$hourOfDay < 22,
                                           3, # Evening
                                              4 ))) # Night

# Seq vector to help shift 'count' rows by one hour to analysis predicability
ind <- seq(2, nrow(bikes) + 1, 1)
ind[length(ind)] <- NA

# 19, 20 Add column showing the 'next hour" datetime and count
bikes$nextHourDateTime <- bikes[ind,"datetime"]
bikes$nextHourCount <- bikes[ind, "count"]
```

### Summary Statistics

Using the `summary()` R function, the basic statistics about each attribute are summarized through the raw R output that follows.

```r
# Summary
summary(bikes)
```

```
##                 datetime        season       holiday
##  2011-01-01 00:00:00:    1   Min.   :1.00   Min.   :0.0000
##  2011-01-01 01:00:00:    1   1st Qu.:2.00   1st Qu.:0.0000
##  2011-01-01 02:00:00:    1   Median :3.00   Median :0.0000
##  2011-01-01 03:00:00:    1   Mean   :2.51   Mean   :0.0286
##  2011-01-01 04:00:00:    1   3rd Qu.:4.00   3rd Qu.:0.0000
##  2011-01-01 05:00:00:    1   Max.   :4.00   Max.   :1.0000
##  (Other)            :10880
##    workingday       weather          temp           atemp
##  Min.   :0.000   Min.   :1.00   Min.   : 0.82   Min.   : 0.76
##  1st Qu.:0.000   1st Qu.:1.00   1st Qu.:13.94   1st Qu.:16.66
##  Median :1.000   Median :1.00   Median :20.50   Median :24.24
##  Mean   :0.681   Mean   :1.42   Mean   :20.23   Mean   :23.66
##  3rd Qu.:1.000   3rd Qu.:2.00   3rd Qu.:26.24   3rd Qu.:31.06
##  Max.   :1.000   Max.   :4.00   Max.   :41.00   Max.   :45.45
##
##     humidity       windspeed        casual      registered      count
##  Min.   :  0.0   Min.   : 0.0   Min.   :  0   Min.   :  0   Min.   :  1
##  1st Qu.: 47.0   1st Qu.: 7.0   1st Qu.:  4   1st Qu.: 36   1st Qu.: 42
##  Median : 62.0   Median :13.0   Median : 17   Median :118   Median :145
##  Mean   : 61.9   Mean   :12.8   Mean   : 36   Mean   :156   Mean   :192
##  3rd Qu.: 77.0   3rd Qu.:17.0   3rd Qu.: 49   3rd Qu.:222   3rd Qu.:284
##  Max.   :100.0   Max.   :57.0   Max.   :367   Max.   :886   Max.   :977
##
##   seasonName          hourOfDay        dayOfWeek     dayOfWeekInt
##  Length:10886       Min.   : 0.0   Friday   :1529   Min.   :1
##  Class :character   1st Qu.: 6.0   Monday   :1551   1st Qu.:2
##  Mode  :character   Median :12.0   Saturday :1584   Median :4
##                     Mean   :11.5   Sunday   :1579   Mean   :4
##                     3rd Qu.:18.0   Thursday :1553   3rd Qu.:6
##                     Max.   :23.0   Tuesday  :1539   Max.   :7
##                                    Wednesday:1551
##   monthOfYear     segmentOfDay            nextHourDateTime nextHourCount
##  Min.   : 1.00   Min.   :1.0   2011-01-01 01:00:00:    1   Min.   :  1
##  1st Qu.: 4.00   1st Qu.:1.0   2011-01-01 02:00:00:    1   1st Qu.: 42
##  Median : 7.00   Median :2.0   2011-01-01 03:00:00:    1   Median :145
##  Mean   : 6.52   Mean   :2.5   2011-01-01 04:00:00:    1   Mean   :192
##  3rd Qu.:10.00   3rd Qu.:4.0   2011-01-01 05:00:00:    1   3rd Qu.:284
##  Max.   :12.00   Max.   :4.0   (Other)            :10880   Max.   :977
##                                NA's               :    1   NA's   :1
```

### Outlier Analysis

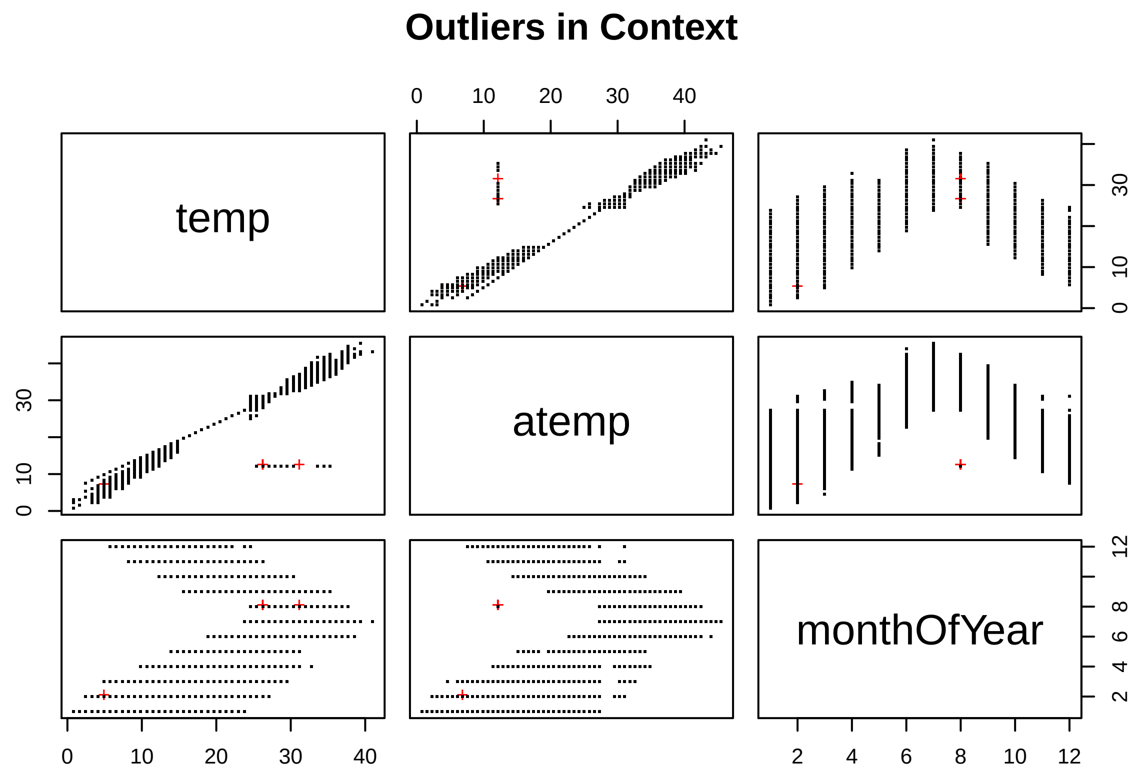
Using the Data Mining with R (DMwR) package's `lofactor()` function, an outlier analysis was performed. Although a complete analysis was performed across all categorical and numeric attributes, generally no extreme outliers were detected, with a notable exception. The following R code illustrates the approach used in the outlier analysis with the top 5 outlying points highlighted in the following charts via a red plus symbol.

```
outlier.scores <- DMwR::lofactor(bikes[,c(2:9,12)], k=5)
```

```
## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009
```

```
outliers <- order(outlier.scores, decreasing=T)[1:5]
```

One might reasonably expect temperature and apparent temperature to follow one another more or less linearly. For each unit increase in temperature, apparent temperature would increase by approximately one unit. As shown in the following chart, this is mostly true, except for the August 17, 2012 values. On this day, the connection between temperature and apparent temperature appear broken with `atemp` stuck at 12.12 °C. Without further knowledge of the data's origins, one can only speculate as to why this is the case.



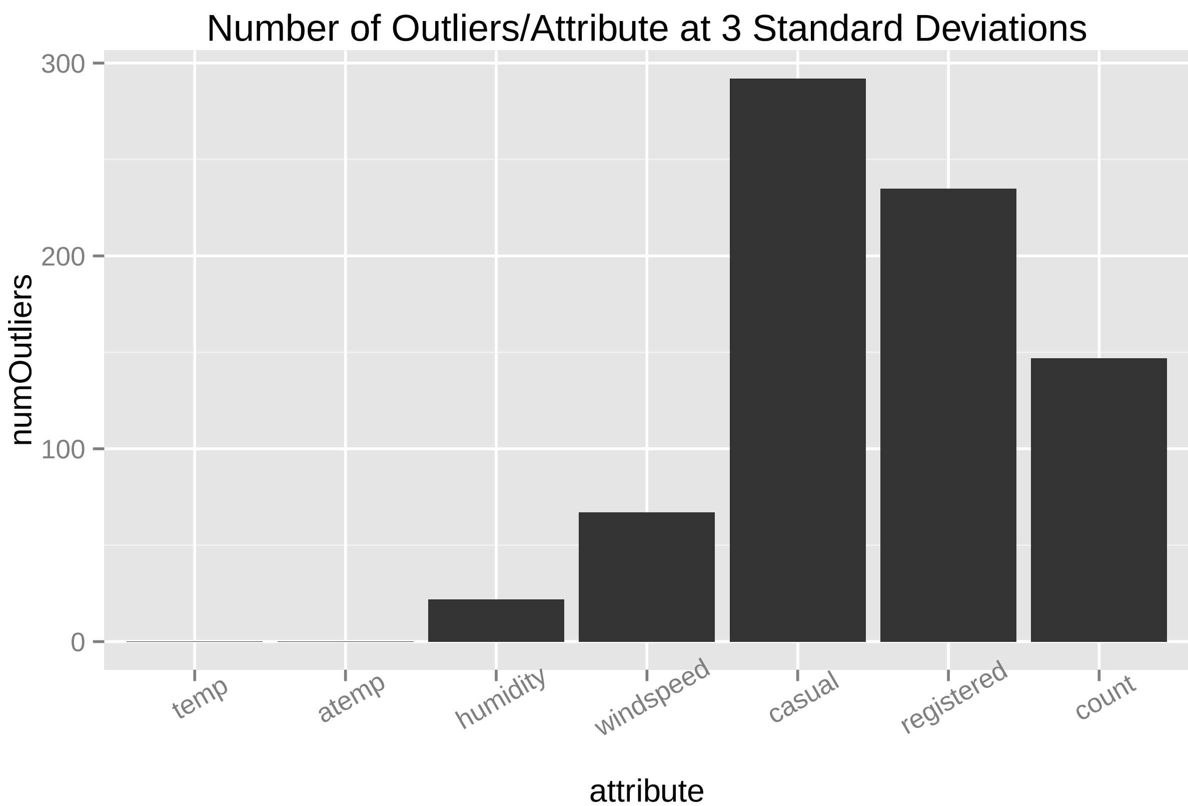

Using an alternative approach to outlier detection, the mean and standard deviation were calculated for the numeric attributes. A distribution analysis was not performed, and as such normal distribution is only assumed here. With a 3 standard deviation width on either side of the mean, values that appear outside these bounds could be considered outliers. The following R code performs this analysis and shows the results.

```
# Calculate mean/standard deviation
msd <- sapply(bikes[,c(6:12)],
              function(cl) c(mean=mean(cl,na.rm=TRUE),
                             stdev=sd(cl,na.rm=TRUE)))
```

```r
# Melt into long form
msdDf <- as.data.frame(t(msd))
msdDf <- data.frame(attribute=rownames(msdDf), msdDf)
msdDf <- subset(msdDf, !is.na(msdDf$mean) & !is.na(msdDf$stdev))

# Add lower/upper bounds at 3 stdevs
xTimes <- 3
lowers <- msdDf$mean - (xTimes * msdDf$stdev)
uppers <- msdDf$mean + (xTimes * msdDf$stdev)
msdDf <- data.frame(msdDf, lower=lowers, upper = uppers)
msdDf
```

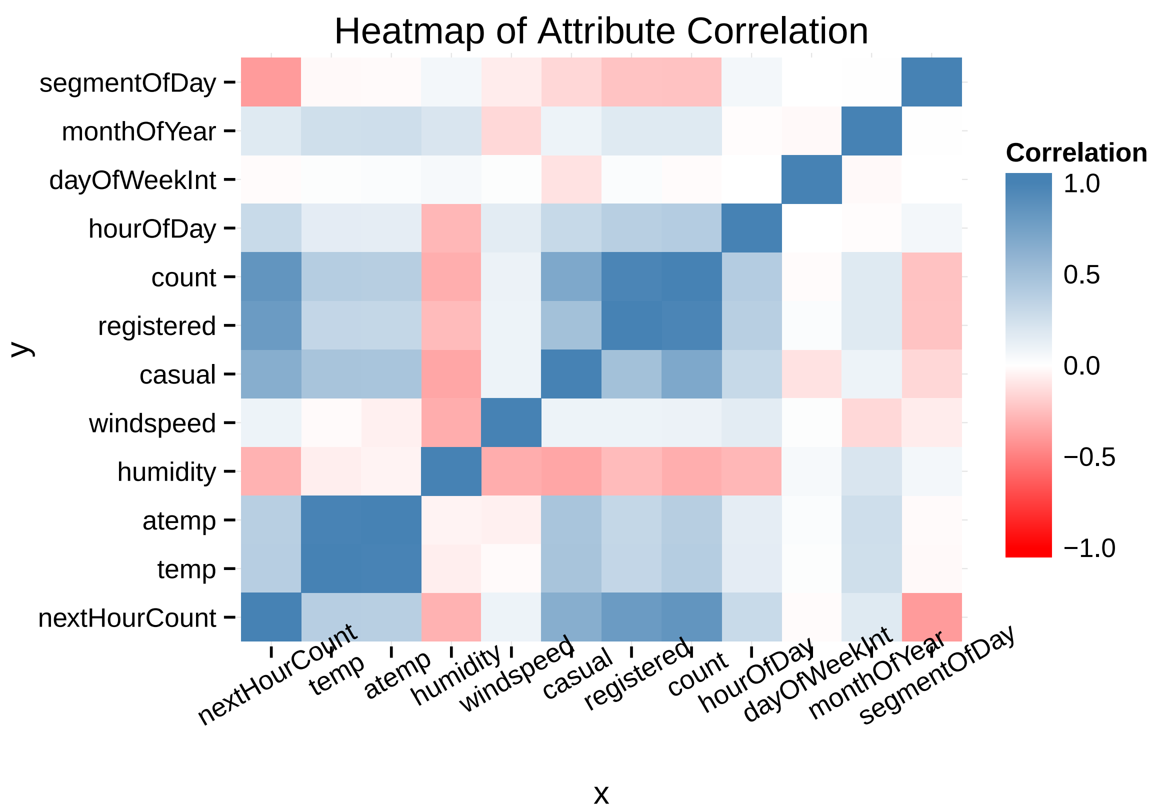
```
##             attribute   mean   stdev    lower  upper
## temp             temp  20.23   7.792   -3.144  43.61
## atemp           atemp  23.66   8.475   -1.769  49.08
## humidity     humidity  61.89  19.245    4.151 119.62
## windspeed   windspeed  12.80   8.165  -11.694  37.29
## casual         casual  36.02  49.960 -113.859 185.90
## registered registered 155.55 151.039 -297.565 608.67
## count           count 191.57 181.144 -351.859 735.01
```




### Correlation Analysis

Using R's `cor()` function, as shown in the following code, an analysis of correlation between the numeric attributes was performed.

```r
bikeCor <- cor(bikes[, c(20,6:12,14,16,17,18)], use="complete.obs")
bikesCorMelt <- reshape2::melt(bikeCor, varnames=c("x", "y"), value.name="Correlation")
bikesCorMelt <- bikesCorMelt[order(bikesCorMelt$Correlation),]
```




As can be seen in the heat map above and values below, humidity is negatively correlated with bike sharing demand across all three measures (casual, registered and count). Likewise, temperature is positively correlated with bike sharing demand in this data set. Interestingly, the day of the week `dayOfWeekInt` attribute shows virtually no correlation with bike sharing demand, but `segmentOfDay`, the indicator of morning, afternoon, etc, shows a moderate negative correlation whereby apparently more rentals occur in the morning (`segmentOfDay` = 1).

```
##               nextHourCount     temp    atemp humidity windspeed   casual
## nextHourCount       1.00000  0.38427  0.37812 -0.29991   0.09262  0.64575
## temp                0.38427  1.00000  0.98495 -0.06493  -0.01789  0.46707
## atemp               0.37812  0.98495  1.00000 -0.04352  -0.05751  0.46204
## humidity           -0.29991 -0.06493 -0.04352  1.00000  -0.31860 -0.34818
## windspeed           0.09262 -0.01789 -0.05751 -0.31860   1.00000  0.09225
## casual              0.64575  0.46707  0.46204 -0.34818   0.09225  1.00000
## registered          0.79623  0.31855  0.31461 -0.26545   0.09103  0.49724
## count               0.84201  0.39443  0.38976 -0.31737   0.10135  0.69040
## hourOfDay           0.29200  0.14559  0.14049 -0.27808   0.14672  0.30219
## dayOfWeekInt       -0.01246  0.01657  0.02323  0.04646   0.01750 -0.11358
## monthOfYear         0.16695  0.25776  0.26433  0.20453  -0.15014  0.09283
## segmentOfDay       -0.39285 -0.01899 -0.01514  0.06511  -0.07025 -0.15683
##               registered    count hourOfDay dayOfWeekInt monthOfYear
## nextHourCount    0.79623  0.84201  0.292001   -0.0124619    0.166946
```

```
## temp               0.31855  0.39443  0.145593    0.0165711    0.257762
## atemp              0.31461  0.38976  0.140490    0.0232296    0.264332
## humidity          -0.26545 -0.31737 -0.278080    0.0464605    0.204530
## windspeed          0.09103  0.10135  0.146722    0.0175010   -0.150143
## casual             0.49724  0.69040  0.302187   -0.1135784    0.092828
## registered         1.00000  0.97095  0.380664    0.0240232    0.169542
## count              0.97095  1.00000  0.400745   -0.0112946    0.166968
## hourOfDay          0.38066  0.40074  1.000000    0.0015249   -0.007061
## dayOfWeekInt       0.02402 -0.01129  0.001525    1.0000000   -0.018559
## monthOfYear        0.16954  0.16697 -0.007061   -0.0185592    1.000000
## segmentOfDay      -0.23260 -0.23720  0.062034   -0.0006968    0.004220
##                 segmentOfDay
## nextHourCount     -0.3928521
## temp              -0.0189920
## atemp             -0.0151434
## humidity           0.0651113
## windspeed         -0.0702514
## casual            -0.1568344
## registered        -0.2325953
## count             -0.2371954
## hourOfDay          0.0620341
## dayOfWeekInt      -0.0006968
## monthOfYear        0.0042197
## segmentOfDay       1.0000000
```

### Entropy Analysis

Using Entropy and Information Gain functions developed in a prior exercise, an entropy analysis was performed. Raw entropy of the bike shares per hour was calculated initially.

```r
source(file.path(projRoot, "EntropyFunctions.R"), chdir=TRUE)

# Raw Entropy: Total Bike Sharing
entropy(bikes$nextHourCount)
```

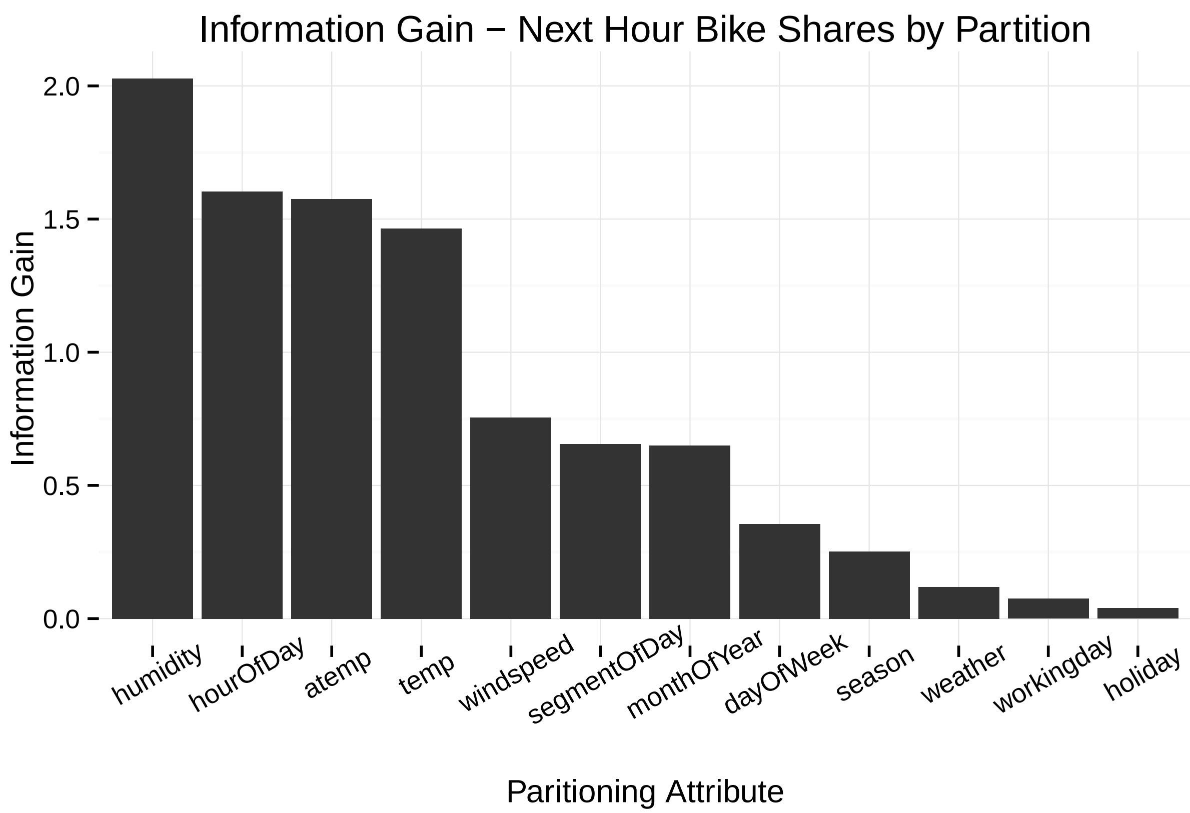
```
## [1] 8.877
```

The `decide()` function from the EntropyFunctions script was used to calculate information gain across all attributes versus the `nextHourCount` bike sharing measure which was added to aid with prediction analysis.

The results were then melted into a long format and sorted for better visualization. The R code is shown below. None of the attributes produced particularly staggering information gain, but the `humidity` attribute was found to be the most meaningful, followed by the `hourOfDay` calculated attribute and `atemp`/`temp` attributes.

```r
# Calculate information gain across all categorical and numeric attributes.
nextHrEnt <- decide(bikes[,c(2,3,4,5,6,7,8,9,14,15,17,18,20)], 13)

nextHrEntMelt <- reshape2::melt(nextHrEnt$gains, value.name="info.gain")
nextHrEntMelt <- cbind(nextHrEntMelt, attribute=rownames(nextHrEntMelt))
nextHrEntMelt <- nextHrEntMelt[order(-nextHrEntMelt$info.gain),]
```




```
##               info.gain    attribute
## humidity        2.02810     humidity
## hourOfDay       1.60364    hourOfDay
## atemp           1.57577        atemp
## temp            1.46479         temp
## windspeed       0.75502    windspeed
## segmentOfDay    0.65582 segmentOfDay
## monthOfYear     0.65021  monthOfYear
## dayOfWeek       0.35485    dayOfWeek
## season          0.25230       season
## weather         0.11851      weather
## workingday      0.07422   workingday
## holiday         0.03910      holiday
```

### Source Code

The raw R markdown code used to produce this data set profile can be found on GitHub, in my DataAcqMgmt repository.

### References

Fanaee-T, Hadi, and Gama, Joao, Event labeling combining ensemble detectors and background knowledge, Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg.

Lander, Jared P. “Correlation and Covariance.” R for Everyone: Advanced Analytics and Graphics. New York: Addison-Wesley, 2014. N. pag. Print.

“Means and SD for Columns in a Dataframe with NA Values.” R. StackOverflow, 27 Dec. 2013. Web. 28 Sept. 2014.

Zhao, Yanchang. “Outlier Detection - RDataMining.com: R and Data Mining” RDataMining.com: R and Data Mining. N.p., 2014. Web. 27 Sept. 2014.