DataSetProfile

Charley Ferrari

Thursday, September 25, 2014

This summary will cover the sample bike share data from kaggle.com, located below:

http://www.kaggle.com/c/bike-sharing-demand

```
library(ggplot2)
#library(xts)

#library(TTR)
#library(forecast)
library(reshape2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)

setwd("C:/Users/Charley/Downloads/Courses/CUNY/SPS/Git/IS 607 Data Acquisition and Management/Project 2
sourcefile <- "Train.csv"

variables <- read.csv(sourcefile, header = TRUE)


variables$datetime <- strptime(variables$datetime, format = "%Y-%m-%d %H:%M:%S", tz="")
```

Listing of attributes The variables are as follows:

datetime - Datetime data, date + timestamp at an hourly frequency

season - Categorical data coded to numbers 1 - 4 1 = spring 2 = summer 3 = fall 4 = winter

holiday - Dummy Variable (1 or 0) whether the day is considered a holiday

workingday - Dummy Variable (1 or 0) whether the day is neither a weekend nor holiday

weather - Categorical data coded to numbers 1 - 4 (in increasing severity of weather) 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 - Numerical variable, temperature in Celsius atemp - Numerical variable, “feels like” temperature in Celsius

humidity - Numerical variable, relative humidity (as a percentage)

windspeed - Numerical variable, wind speed

casual - Numerical variable, number of non-registered user rentals initiated

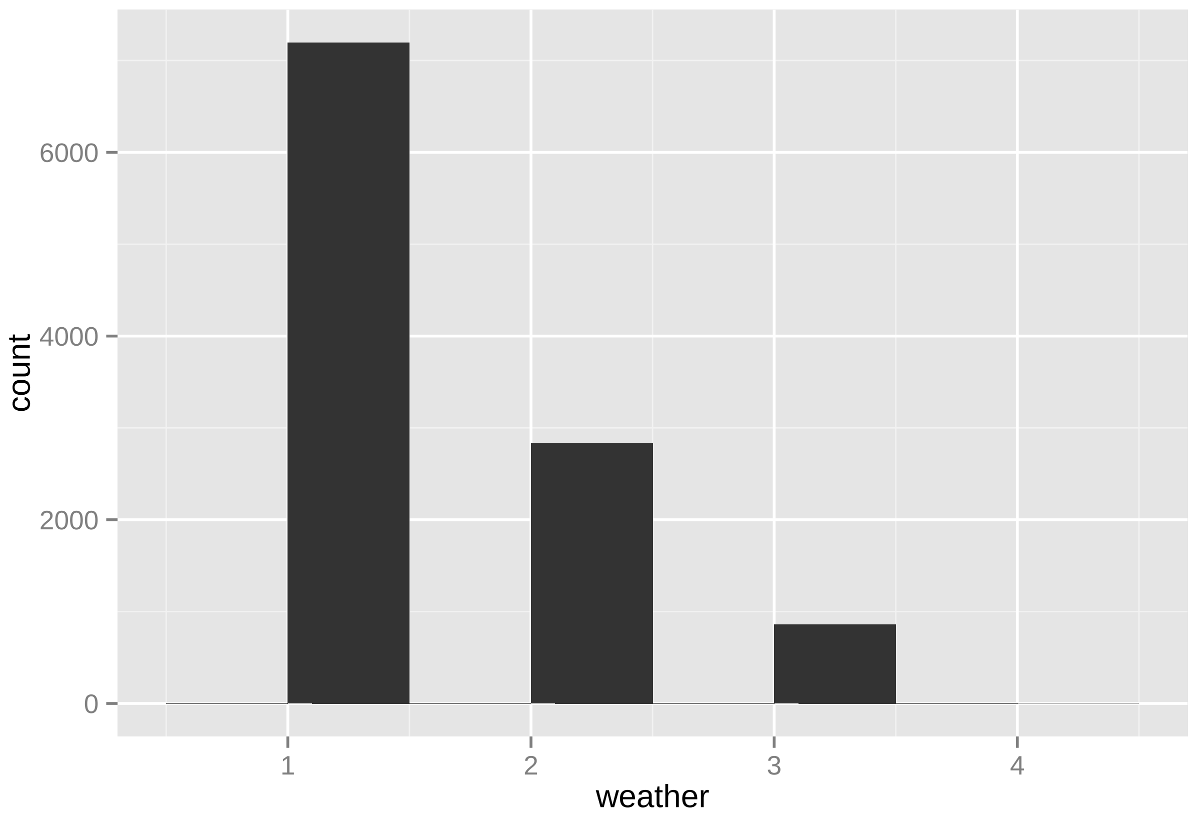
registered - Numerical variable, number of registered user rentals initiated

count - Numerical variable, number of total rentals (sum of casual and registered)

Summaries of relevant variables Dummy and datetime variables won’t be described. Seasons, holidays, and working days vary pretty predictibly. Weather is a dummmy variable, but a histogram should be able to give a rough idea of what the weather is like during the measured time.

```r
ggplot(variables, aes(x=weather)) +
  geom_histogram() +
  stat_bin(binwidth=0.5)
```

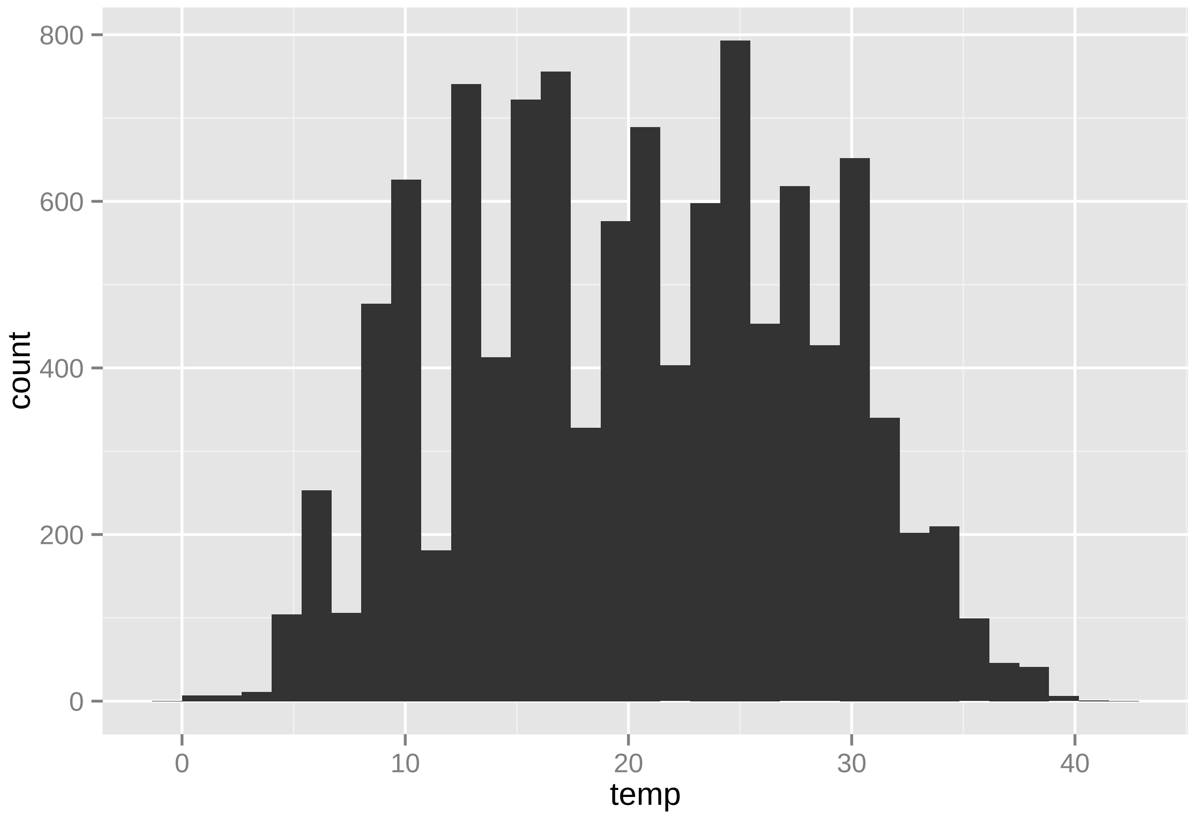
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



There is one observation of 4 which is not really visible

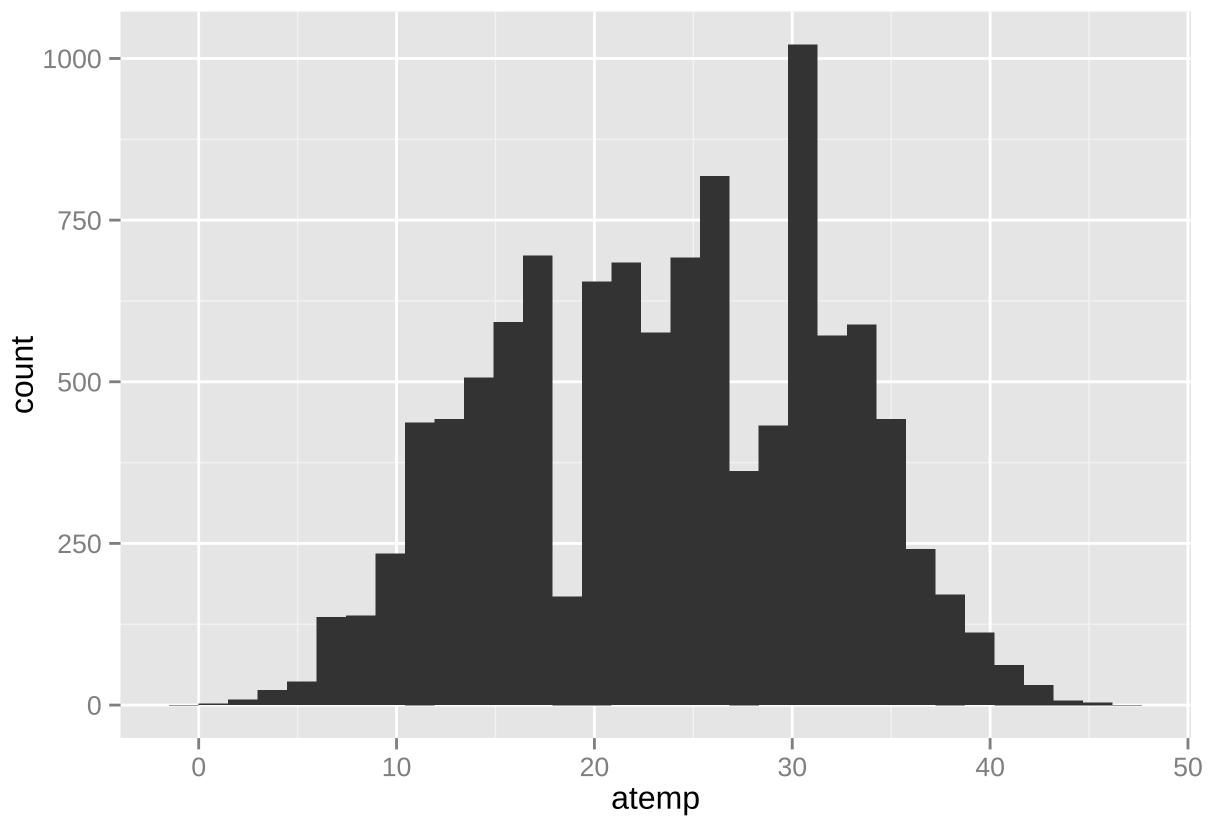
```r
ggplot(variables, aes(x=temp)) +
  geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



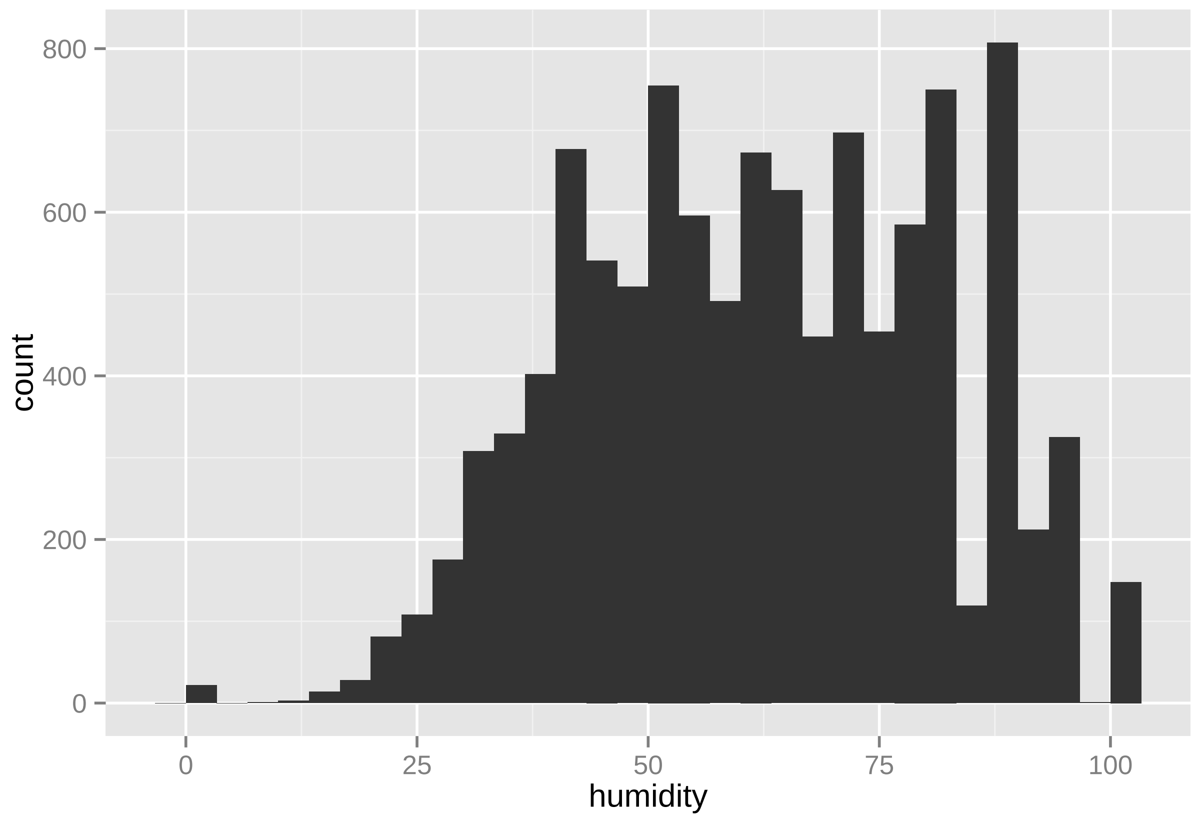
```r
ggplot(variables, aes(x=atemp)) +
  geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



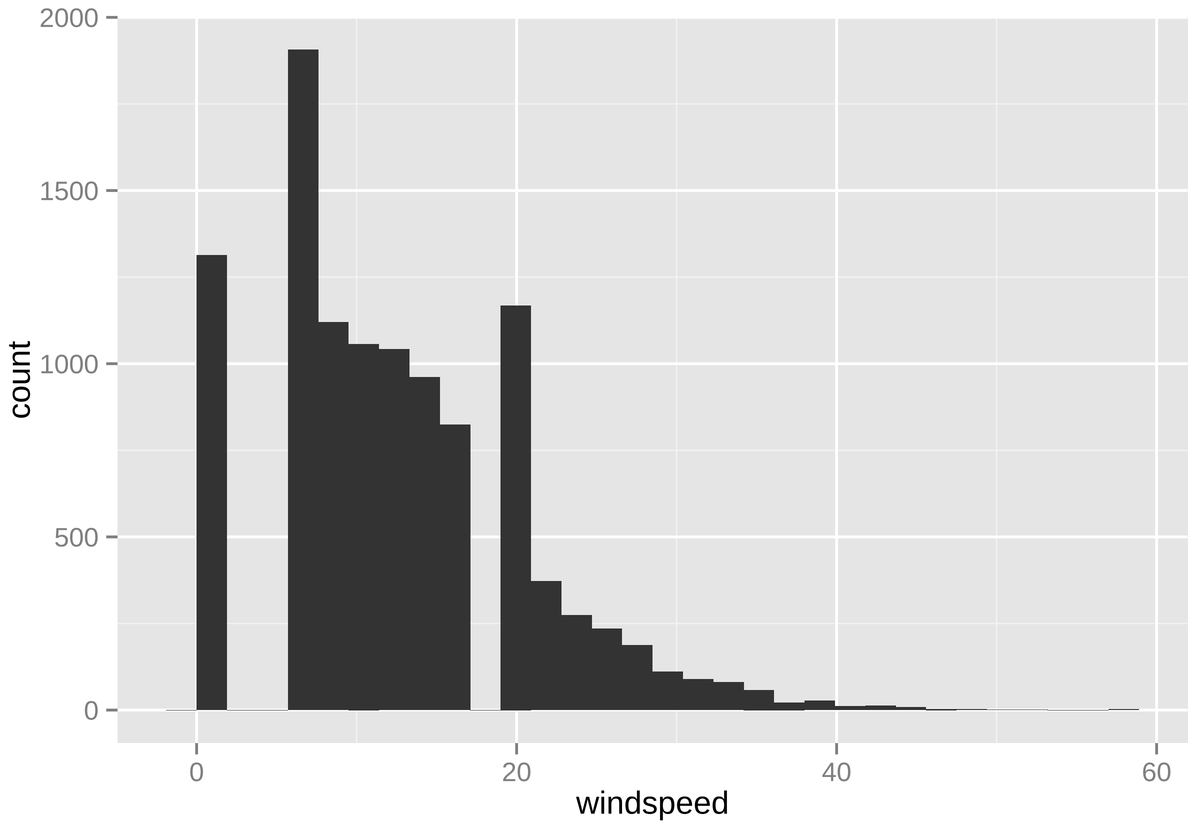
```
ggplot(variables, aes(x=humidity)) +
  geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



```
ggplot(variables, aes(x=windspeed)) +
  geom_histogram()
```

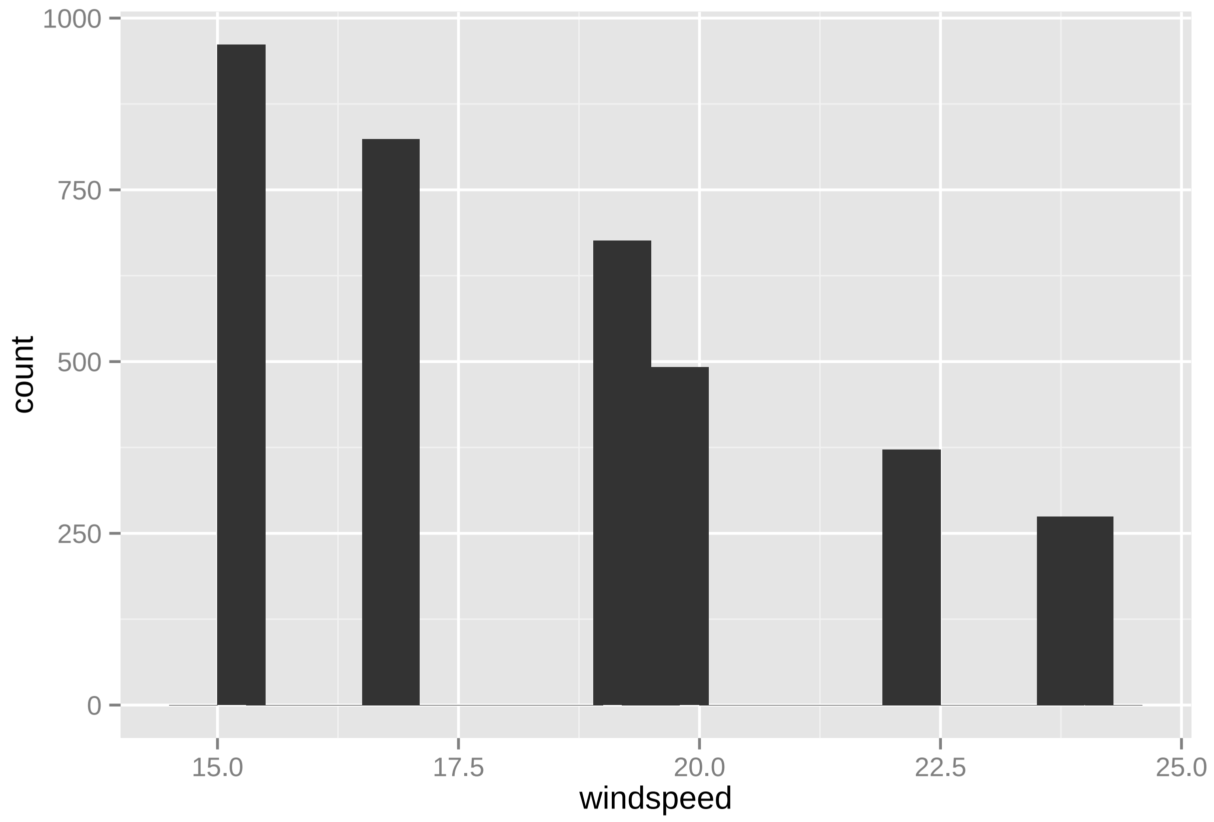
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



These three variables give a weather profile of Washington DC. I'm curious of the spike of wind speed occurrences. There seems to be a weird spike at 20 degrees, and an empty observation before it.

```
# I tried to do this with dplyr but was unable to get ranges to work with it!
# Check the forum for the specifics of the issue I was having.
#variablesws <- filter(variables, windspeed < 25)

variablesws <- variables[(variables$windspeed > 15 & variables$windspeed < 25),]

ggplot(variablesws, aes(x=windspeed)) +
  geom_histogram() +
  stat_bin(binwidth=0.5)
```

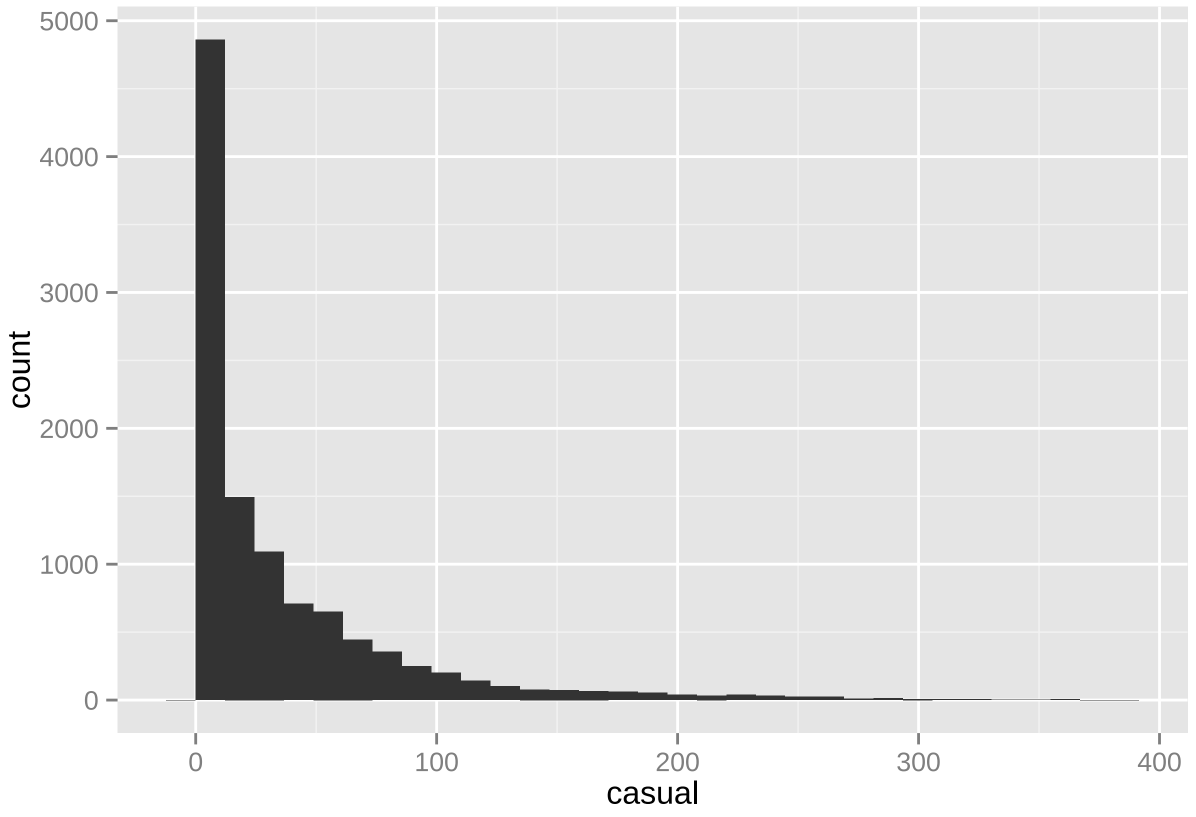
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



The histogram still looks weird when I try to draw it like this. Spot checking the data, I noticed I was unable to find any windspeed between 17.0 and 17.999… Seems like there might be some sort of measurement error occurring here.

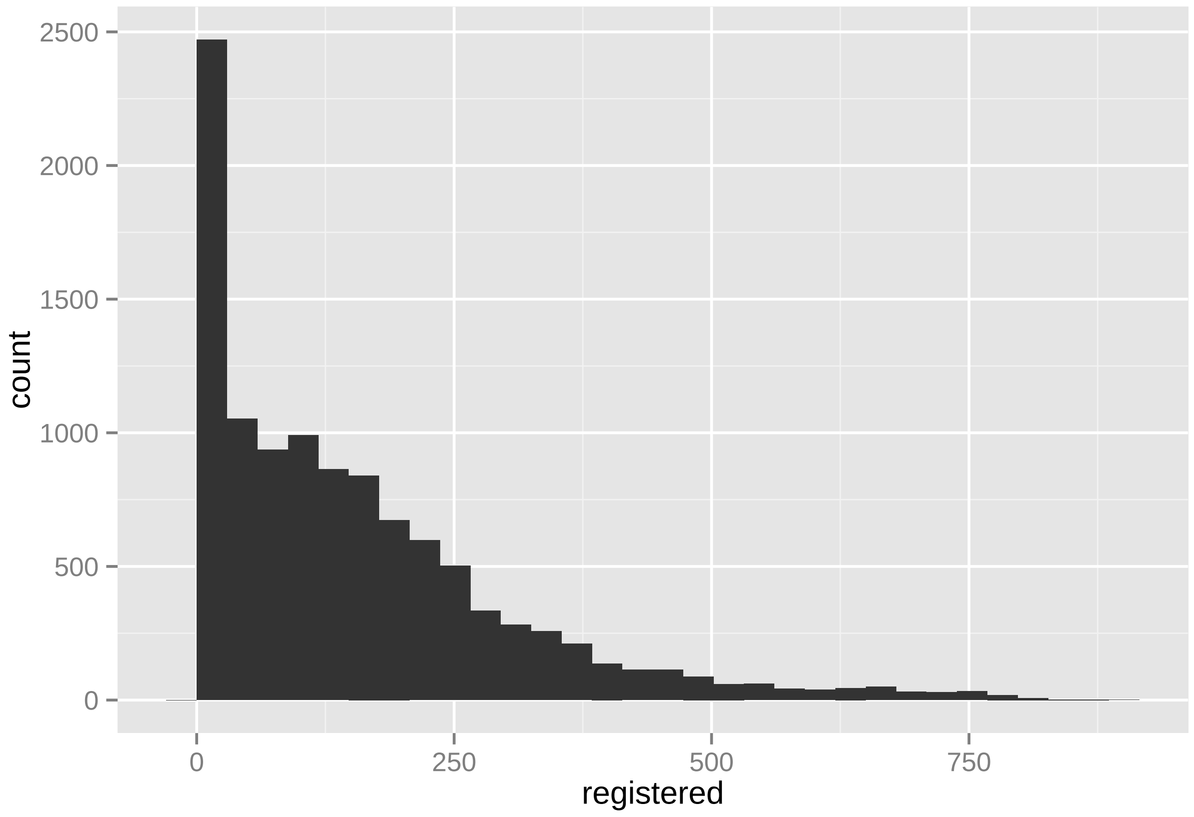
```r
ggplot(variables, aes(x=casual)) +
  geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



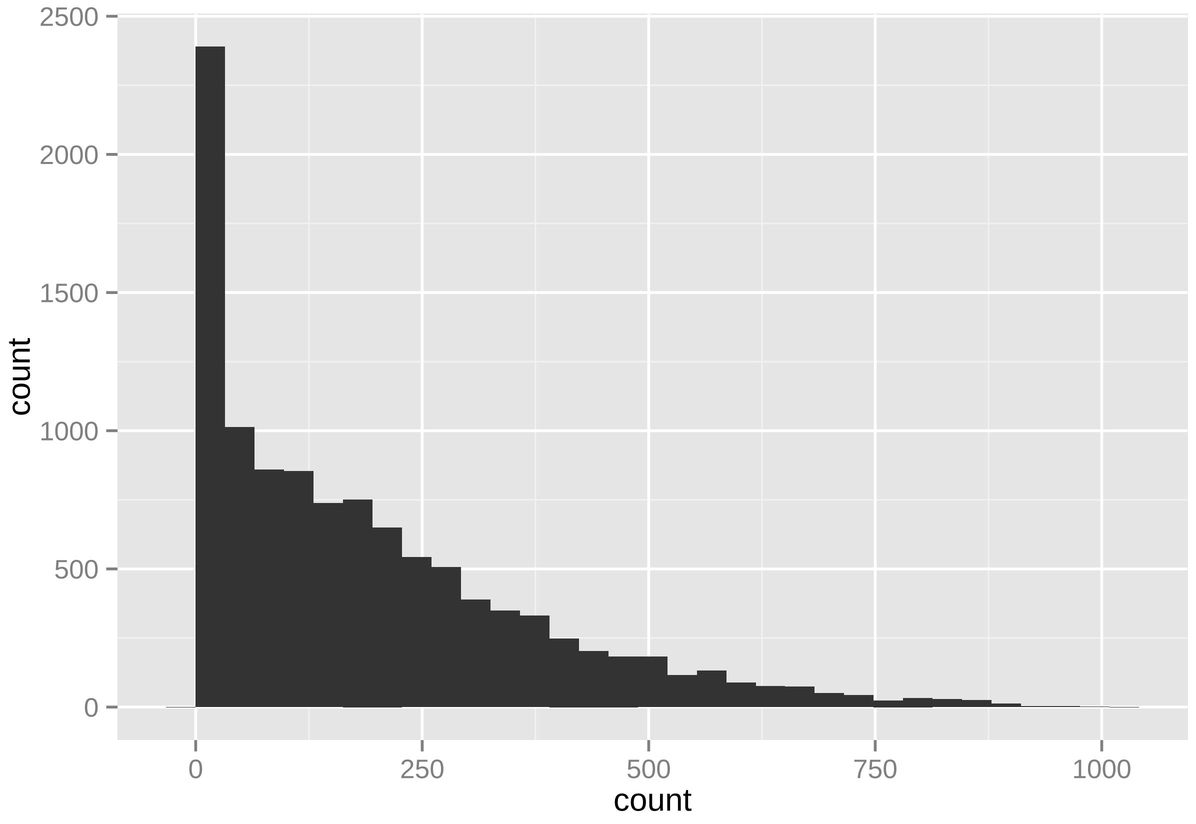
```
ggplot(variables, aes(x=registered)) +
  geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



```
ggplot(variables, aes(x=count)) +
  geom_histogram()
```

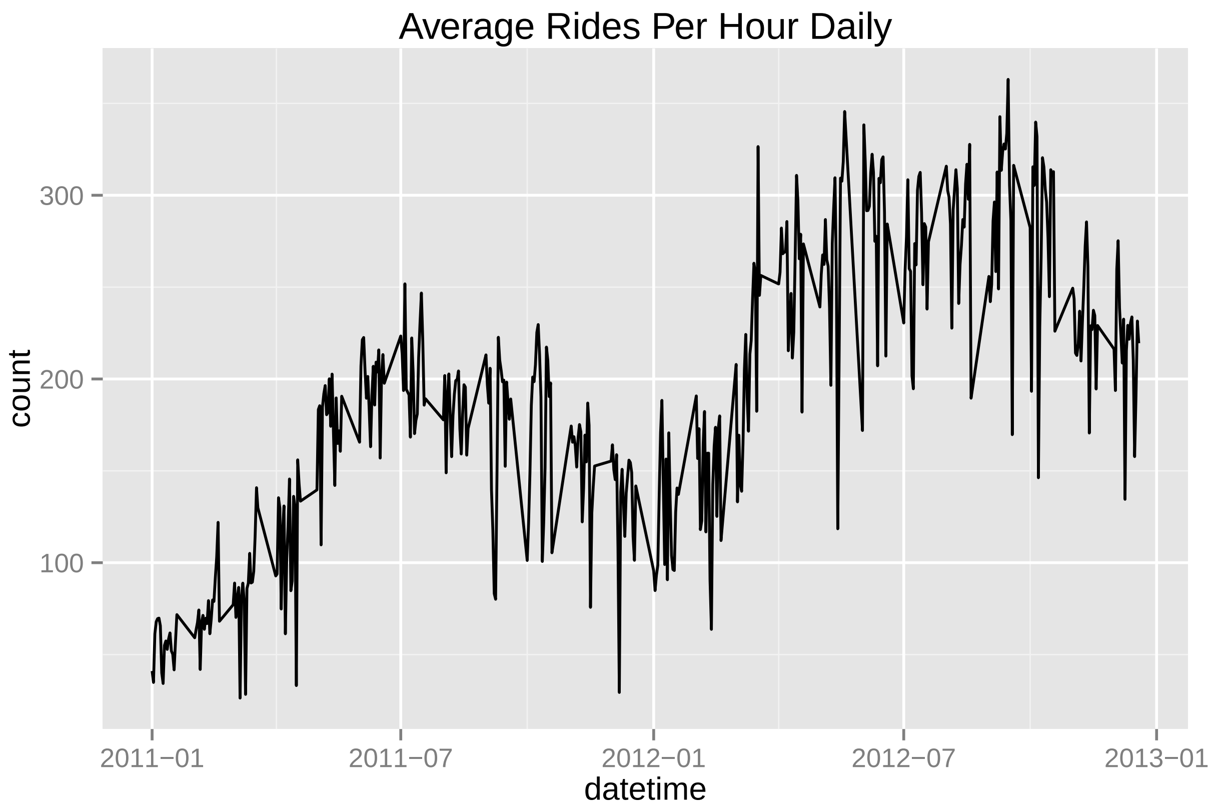
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



The histograms of casual vs. regular riders tell an interesting story. They follow the same general trend of a few busy periods and a tail of less frequent periods. For registered users, there appears to be a plateau, which makes sense given the rush hour pattern of registered users.

Looking at ride counts over time First I'll create a daily dataframe for easier analysis

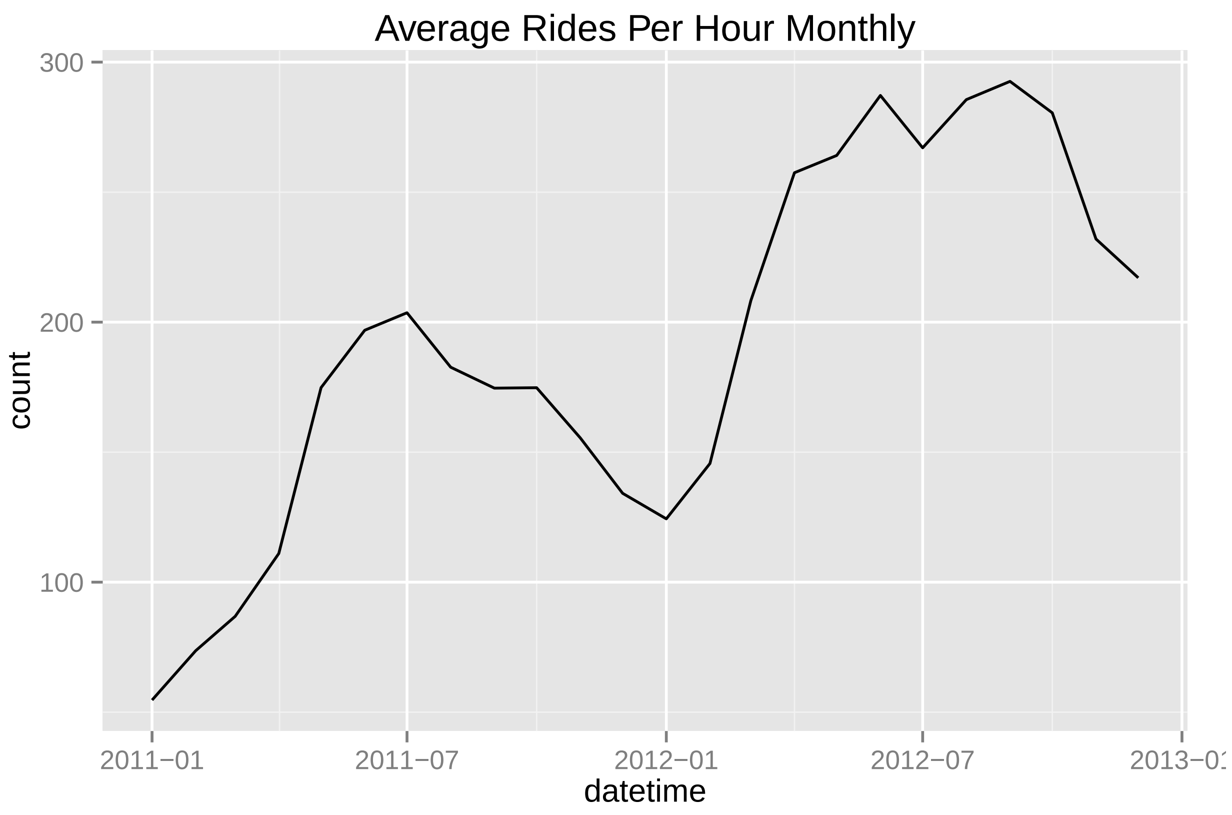
```
variables$Year <- factor(variables$datetime$year + 1900)
variables$Month <- factor(variables$datetime$mon + 1)
variables$Day <- factor(variables$datetime$mday)
variables$Hour <- factor(variables$datetime$hour)

variablesWithDates <- variables
variables <- variables[,colnames(variables) != "datetime"]

# Found out dplyr doesn't like datetime types...

dayvariables <- variables %>%
  group_by(Year, Month, Day, season, holiday, workingday) %>%
  summarise(count = mean(count))

dayvariables$datetime <- strptime(paste(dayvariables$Year, dayvariables$Month, dayvariables$Day, sep=":

ggplot(dayvariables, aes(x=datetime, y=count)) +
  geom_line() +
  ggtitle("Average Rides Per Hour Daily")
```



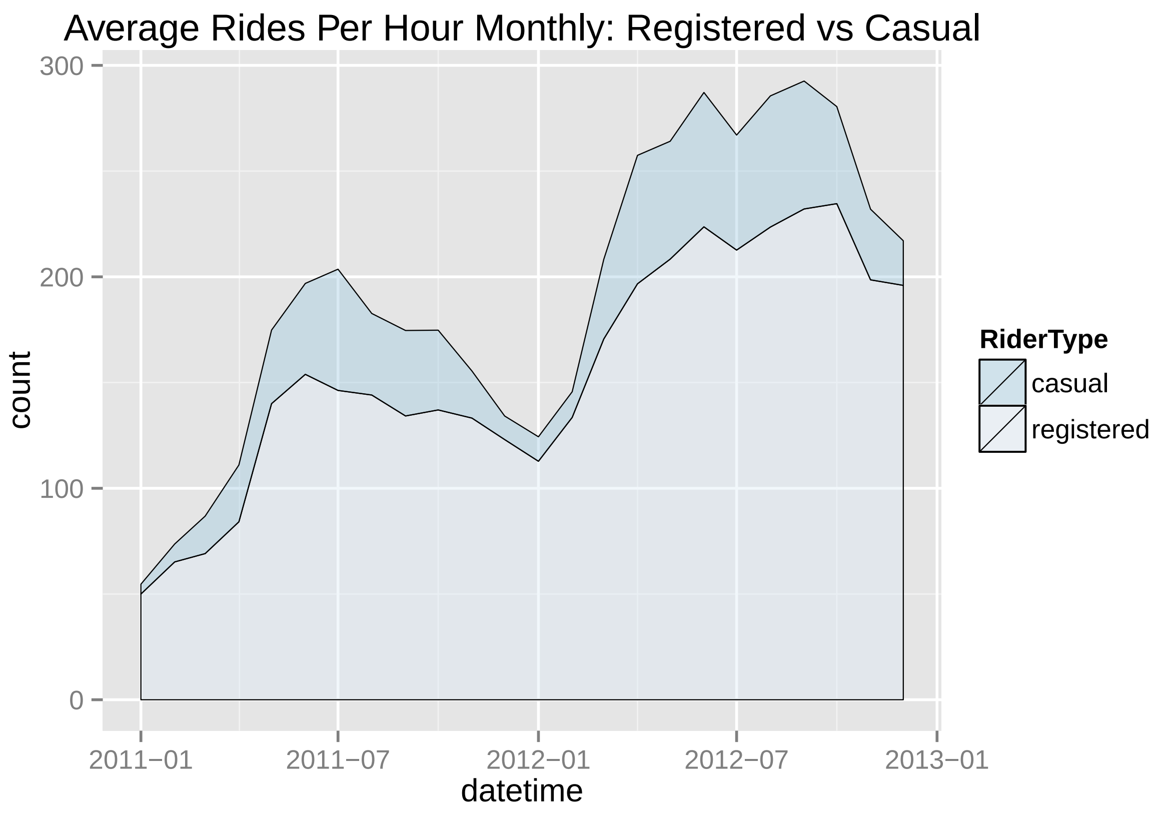
The daily rides data is pretty noisy, so lets try to see what's going on monthly to see a trend.

If this were just time series analysis I'd probably use a moving average, in this case I'm just going to collapse the series to get a similar result.

```
monthvariables <- variables %>%
  group_by(Year, Month) %>%
  summarise(count = mean(count))

monthvariables$datetime <- strptime(paste(monthvariables$Year, monthvariables$Month, "1", sep=":"), for

ggplot(monthvariables, aes(x=datetime, y=count)) +
  geom_line() +
  ggtitle("Average Rides Per Hour Monthly")
```



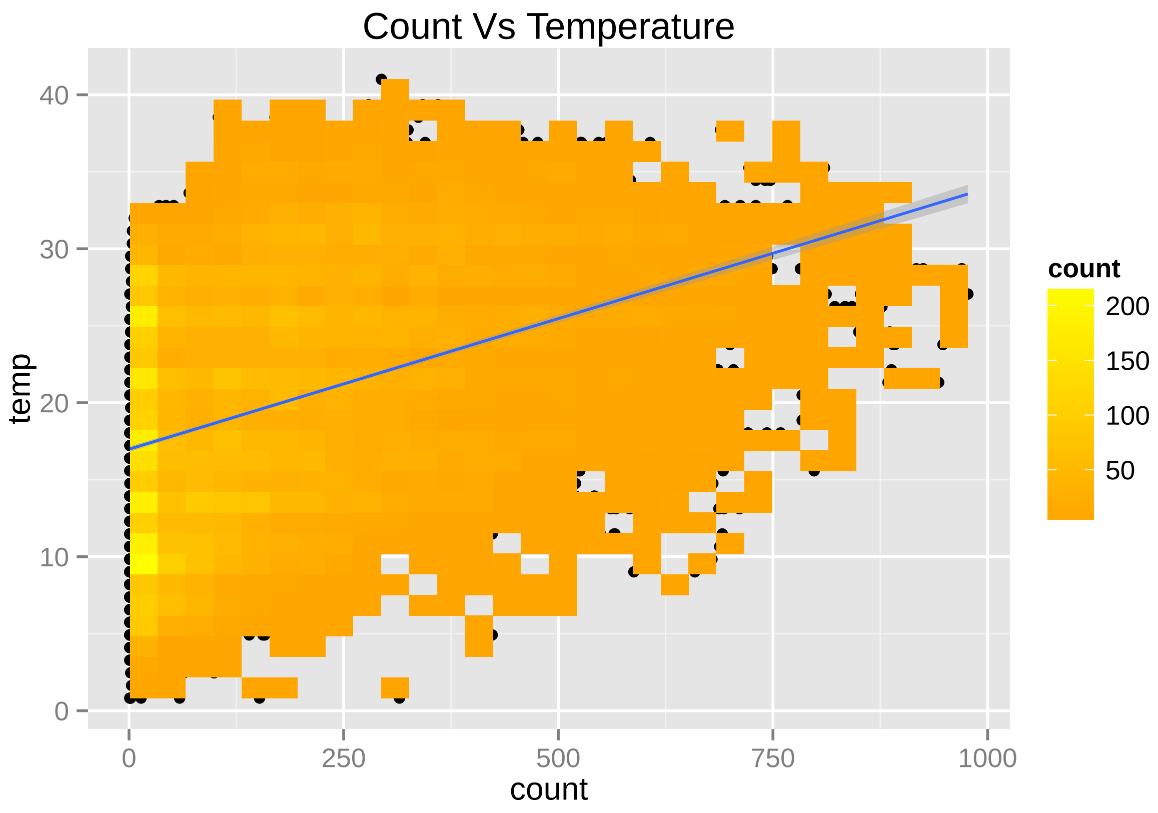
```
registered <- variables %>%
  group_by(Year, Month) %>%
  summarise(registered = mean(registered))

casual <- variables %>%
  group_by(Year, Month) %>%
  summarise(casual = mean(casual))

monthvariables <- cbind(monthvariables, registered$registered, casual$casual)

rm(casual, registered)

colnames(monthvariables) <- c("Year", "Month", "count", "datetime", "registered", "casual")

monthvariablesstacked <- monthvariables %>% select(datetime, registered, casual)

monthvariablesstacked$datetime <- as.character(monthvariablesstacked$datetime)

monthvariablesstacked <- monthvariablesstacked %>% gather(RiderType, count, registered:casual)

monthvariablesstacked$datetime <- strptime(monthvariablesstacked$datetime, format = "%Y-%m-%d", tz="")

monthvariablesstacked$RiderType <- factor(monthvariablesstacked$RiderType)

ggplot(monthvariablesstacked, aes(x=datetime, y=count, fill=RiderType)) +
  geom_area(colour="black", size=.2, alpha=.4) +
  scale_fill_brewer(palette="Blues", breaks=rev(levels(monthvariablesstacked$RiderType))) +
  ggtitle("Average Rides Per Hour Monthly: Registered vs Casual")
```



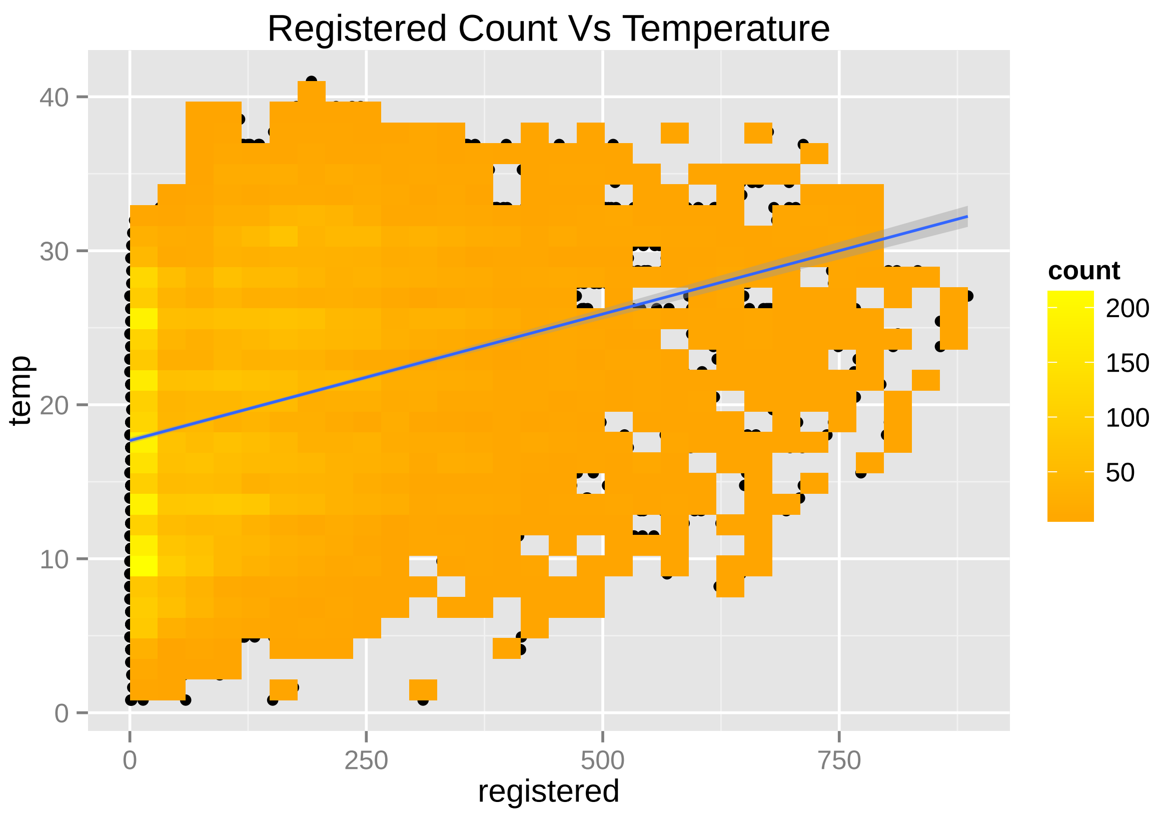
Here we see the more general trend. Ridership seems to be increasing overall, with seasonal increases during the warmer months. There appears to be a large amount of registered riders as opposed to casual riders. Predictibly, a greater portion of the total riders are casual during the summer months, suggesting that the casual riders are more sensitive to changes in weather.

How Weather Affects Ridership We can get some broad pictures of this using scatter plots. Below is an example of different types of count and temperature:

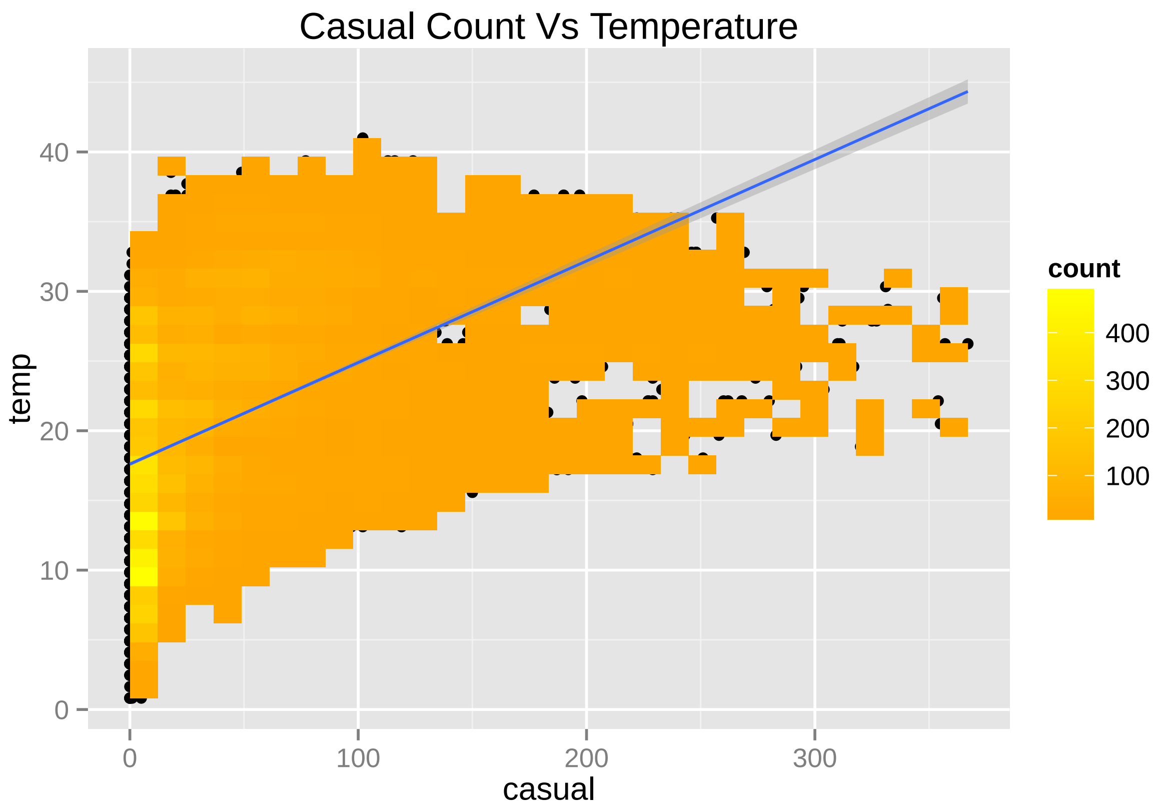
```
ggplot(variables, aes(x=count, y=temp)) +
  geom_point() +
  stat_bin2d() +
  scale_fill_gradient(low="orange", high="yellow") +
  stat_smooth(method=lm) +
  ggtitle("Count Vs Temperature")
```



```r
ggplot(variables, aes(x=registered, y=temp)) +
  geom_point() +
  stat_bin2d() +
  scale_fill_gradient(low="orange", high="yellow") +
  stat_smooth(method=lm) +
  ggtitle("Registered Count Vs Temperature")
```



```
ggplot(variables, aes(x=casual, y=temp)) +
  geom_point() +
  stat_bin2d() +
  scale_fill_gradient(low="orange", high="yellow") +
  stat_smooth(method=lm) +
  ggtitle("Casual Count Vs Temperature")
```



This can give us some interesting views, for instance, casual users seem to be a bit more affected by temperature than registered users, which makes a bit of sense considering the nature of the rider.

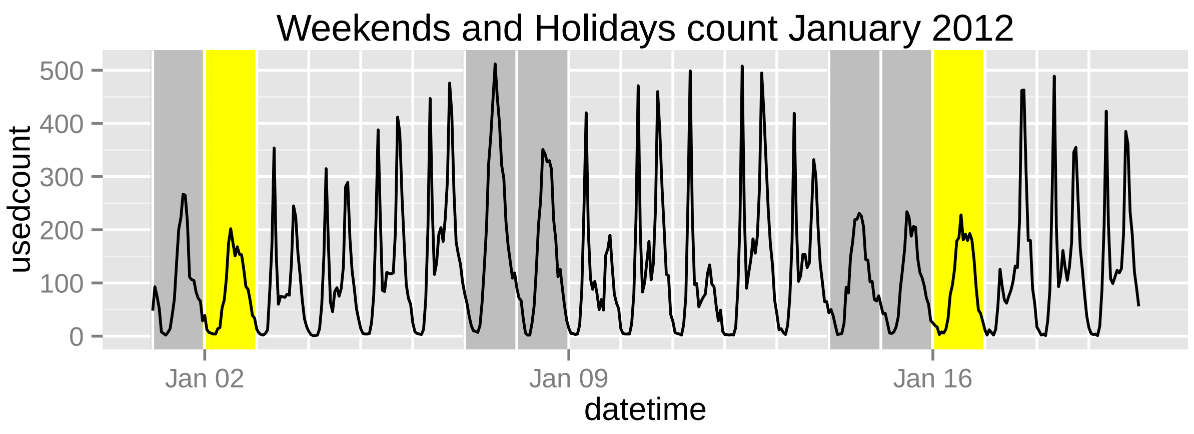
Because this is time series data however, it's better to view the data in that context.

Using an example month we'll try to get an idea of how weather is affecting ridership.

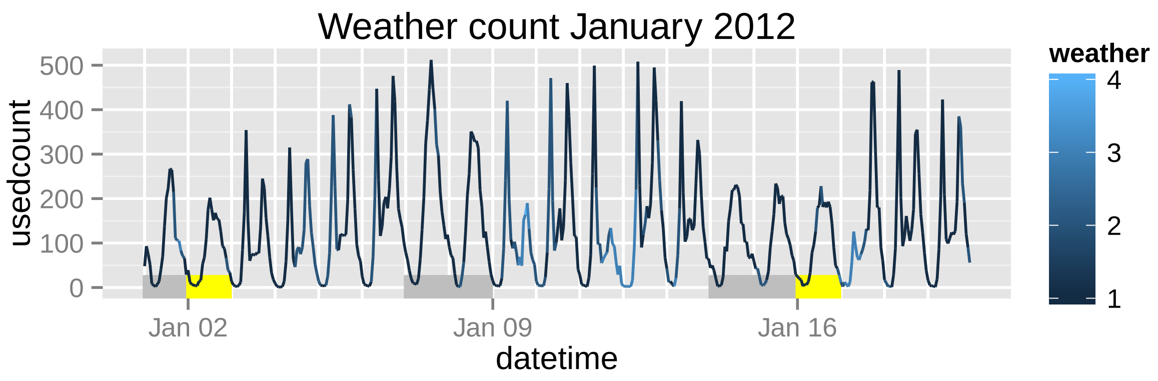
I originally wanted to present this as a for loop for each month and cycling through count, registered, and casual, but that resulted in a large amount of pages. A single month gets the idea of what is occurring across.

```
m<- 1
y <- 2012
columntitle <- "count"

variables <- variablesWithDates

Months <- c("January", "February", "March", "April", "May", "June",  "July", "August", "September", "Oc

newvariables <- variables[,(!(colnames(variables) == "count") & !(colnames(variables) == "registered") &

newvariables[,"usedcount"] <- variables[,columntitle]

slice <- newvariables[newvariables$Year == y & newvariables$Month == m,]

#slice <- newvariables %>% filter(Year == y, Month == m)
# Not working because of datetime again.

################
```

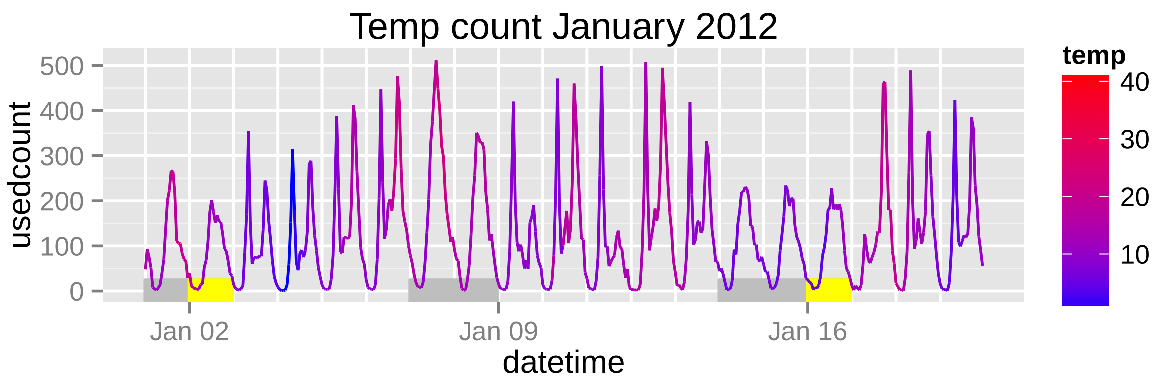
```
# Weekends & Holidays
################

if(NROW(slice[slice$holiday == 1,]) == 0){
  ggplot(slice,aes(x=datetime, y=usedcount)) +
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime - 1000
    #geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = Inf, xmin = datetime - 1000,
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_line() +
    ggtitle(paste("Weekends and Holidays", columntitle, Months[m], y, sep=" ")) +
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}else{
  ggplot(slice,aes(x=datetime, y=usedcount)) +
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime - 1000
    geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = Inf, xmin = datetime - 1000, x
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_line() +
    ggtitle(paste("Weekends and Holidays", columntitle, Months[m], y, sep=" ")) +
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}
```



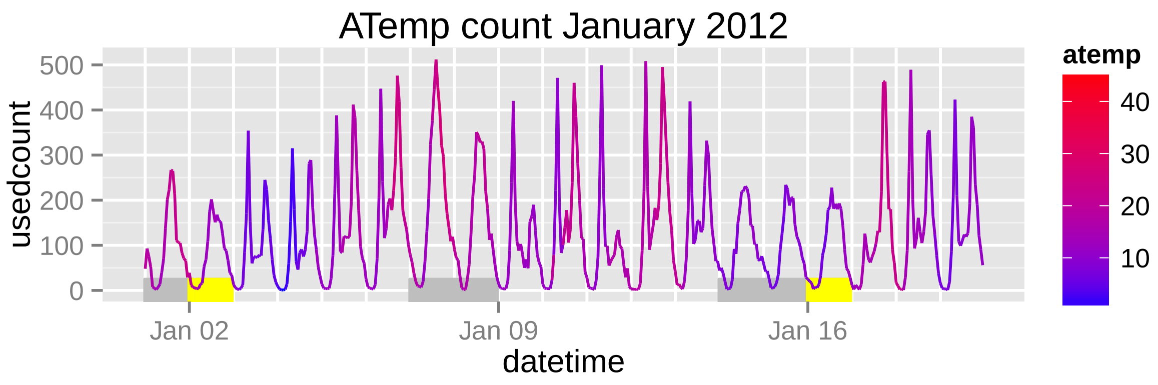
```
################
# Weather
# My original plan with weather was to
# draw rectangles for the weather. This
# ended up looking too busy.
################

if(NROW(slice[slice$holiday == 1,]) == 0){
  #slice$weather = factor(slice$weather, levels=c(1,2,3,4))
  ggplot(slice,aes(x=datetime, y=usedcount, colour=weather)) +
    #geom_rect(data = slice[slice$weather == 1,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax =
    #geom_rect(data = slice[slice$weather == 2,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax =
    #geom_rect(data = slice[slice$weather == 3,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax =
    #geom_rect(data = slice[slice$weather == 4,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax =
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    #geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, x
    geom_line() +
    ggtitle(paste("Weather", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(1, 4)) +
    #scale_colour_manual(values = cols) +
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}else{
  ggplot(slice,aes(x=datetime, y=usedcount, colour=weather)) +
    #geom_rect(data = slice[slice$weather == 1,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax =
    #geom_rect(data = slice[slice$weather == 2,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax =
    #geom_rect(data = slice[slice$weather == 3,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax =
    #geom_rect(data = slice[slice$weather == 4,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax =
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, xm
    geom_line() +
    ggtitle(paste("Weather", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(1, 4)) +
    #scale_colour_manual(values = cols) +
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}
```



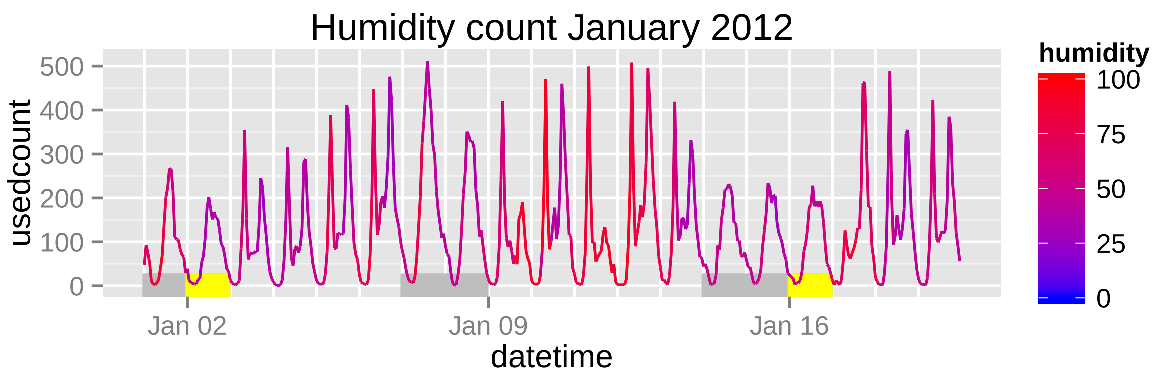
```
################
# Temp
################

if(NROW(slice[slice$holiday == 1,]) == 0){
  ggplot(slice,aes(x=datetime, y=usedcount, colour=temp)) +
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    #geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, x
    geom_line() +
    ggtitle(paste("Temp", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(min(newvariables$temp), max(newvariables$temp)), low="blue", high="r
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}else{
  ggplot(slice,aes(x=datetime, y=usedcount, colour=temp)) +
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, xma
    geom_line() +
    ggtitle(paste("Temp", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(min(newvariables$temp), max(newvariables$temp)), low="blue", high="r
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}
```



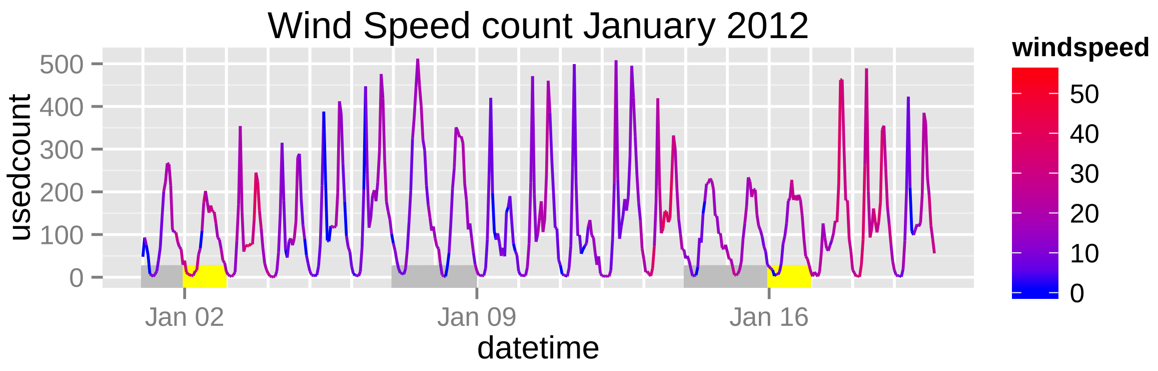
```
################
# ATemp
################

if(NROW(slice[slice$holiday == 1,]) == 0){
  ggplot(slice,aes(x=datetime, y=usedcount, colour=atemp)) +
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    #geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, x
    geom_line() +
    ggtitle(paste("ATemp", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(min(newvariables$atemp), max(newvariables$atemp)), low="blue", high=
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}else{
  ggplot(slice,aes(x=datetime, y=usedcount, colour=atemp)) +
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, xma
    geom_line() +
    ggtitle(paste("ATemp", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(min(newvariables$atemp), max(newvariables$atemp)), low="blue", high=
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}
```



```
################
# Humidity
################

if(NROW(slice[slice$holiday == 1,]) == 0){
  ggplot(slice,aes(x=datetime, y=usedcount, colour=humidity)) +
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    #geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, x
    geom_line() +
    ggtitle(paste("Humidity", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(min(newvariables$humidity), max(newvariables$humidity)), low="blue",
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}else{
  ggplot(slice,aes(x=datetime, y=usedcount, colour=humidity)) +
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, xma
    geom_line() +
    ggtitle(paste("Humidity", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(min(newvariables$humidity), max(newvariables$humidity)), low="blue",
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}
```



```
################
# WindSpeed
################

if(NROW(slice[slice$holiday == 1,]) == 0){
  ggplot(slice,aes(x=datetime, y=usedcount, colour=windspeed)) +
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    #geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, x
    geom_line() +
    ggtitle(paste("Wind Speed", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(min(newvariables$windspeed), max(newvariables$windspeed)), low="blue
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}else{
  ggplot(slice,aes(x=datetime, y=usedcount, colour=windspeed)) +
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$Hour == 0,], aes(ymin = -Inf, ymax = Inf, xmin = datetime, xmax = date
    geom_rect(data = slice[slice$workingday == 0,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000,
    geom_rect(data = slice[slice$holiday == 1,], aes(ymin = -Inf, ymax = 25, xmin = datetime - 1000, xma
    geom_line() +
    ggtitle(paste("Wind Speed", columntitle, Months[m], y, sep=" ")) +
    scale_colour_gradient(limits=c(min(newvariables$windspeed), max(newvariables$windspeed)), low="blue
    coord_fixed(ratio=452000/max(slice[,"usedcount"])) +
    theme(panel.grid.major.x = element_blank(), panel.grid.minor.x = element_blank())
}
```



There are some pretty clear patterns. There are weekday and weekend patterns: weekdays have two peaks (for the rush hours) while weekends and holidays have single peaks. Weather seems to decrease usage a bit, as can be seen in the temperature and weather type graphs.

Count is shown, and registered would behave fairly similarly, but casual riders peak on the weekends and holidays rather than weekdays. This is something that's very clear looking at the graphs over time, but not really clear in scatter plots.