FA17 STAT-S670 EDA | Final Project

| Vinoth Aryan Nagabosshanam

November 27, 2017

# Abstract

Among the various means of transport there are none that are cleaner and economical than the bike (also known as bicycle). We are witnessing the rise of automated bike sharing systems that handle bike rentals and returns with great efficacy. We have with us two years' data (2011 and 2012) from the Capital Bike share system in Washington D.C., USA, which is one such system. Using this data we attempt to build a statistical model that can predict the number of bikes rented at a given hour on a given day using information such as weather, day of the week, whether the day is a holiday etc.

library(ggplot2)

library(GGally)

library(plyr)  
library(grid)  
library(car)  
library(broom)

library(Metrics)

library(randomForest)

library(e1071)

library(MASS)

library(mda)

# Data description

Source: **Kaggle**/UCI Repositor

1. **dteday** : date: the date of the day on which a particular observation was recorded

2. **season** :factor : One of four seasons of the year - Factor

3. **yr, mnth** : factor : attributes extracted from the dteda

4. **hr: factor** :one of twenty four hours of the day

5. **holiday** : factor : whether that particular day was a holiday

6. **weekday** : factor : one of 7 days of the week

7. **workingday** : factor : whether the day was neither a weekend nor a holiday

8. **weathersit** :factor :values from "1" to "4" where "1" indicates clear weather and "4" indicates highly inclement weather

9. **temp** : normalized temperature

10. **atem**p: normalized "feels like" temperature

11. **hum** : normalized humidity

12. **windspeed** : normalized windspeed

13. **cnt** : count of total bikes rented

setwd("C:\\Users\\admin\\Desktop\\EDA\\New folder")  
bike = read.csv("hour\_bike.csv")

# Data at a glance

**Here a quick view of the dataset**

head(bike,n=5)

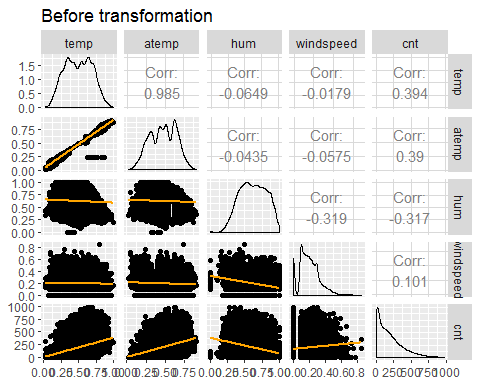
## instant dteday season yr mnth hr holiday weekday workingday  
## 1 1 01-01-2011 1 0 1 0 0 6 0  
## 2 2 01-01-2011 1 0 1 1 0 6 0  
## 3 3 01-01-2011 1 0 1 2 0 6 0  
## 4 4 01-01-2011 1 0 1 3 0 6 0  
## 5 5 01-01-2011 1 0 1 4 0 6 0  
## weathersit temp atemp hum windspeed casual registered cnt type  
## 1 1 0.24 0.2879 0.81 0 3 13 16 train  
## 2 1 0.22 0.2727 0.80 0 8 32 40 train  
## 3 1 0.22 0.2727 0.80 0 5 27 32 train  
## 4 1 0.24 0.2879 0.75 0 3 10 13 train  
## 5 1 0.24 0.2879 0.75 0 0 1 1 train

bike[,c("season","yr","mnth","hr","holiday","weekday","workingday","weathersit")] = lapply(bike[,c("season","yr","mnth","hr","holiday","weekday","workingday","weathersit")],FUN=as.factor)  
bike$dteday = as.Date(bike$dteday,format="%d-%m-%Y")  
bike$std.temp = as.numeric(scale(bike$temp))  
bike$std.atemp = as.numeric(scale(bike$atemp))  
bike$std.hum = as.numeric(scale(bike$hum))  
bike$std.windspeed = as.numeric(scale(sqrt(bike$windspeed)))  
partday = function(x)  
{  
 if(x%in%c("0","1","2","3","4","5","6"))  
 {  
 return("Time-1")  
 }  
 else if(x%in%c("7","8","9","10"))  
 {  
 return("Time-2")  
 }  
 else if(x%in%c("11","12","13","14","15"))  
 {  
 return("Time-3")  
 }  
 else if(x%in%c("16","17","18","19","20"))  
 {  
 return("Time-4")  
 }  
 else  
 {  
 return("Time-5")  
 }  
}  
bike$partday = sapply(bike$hr,partday)  
train = bike[bike$type=="train",-18]

# Univariate/Bivariate Analysis

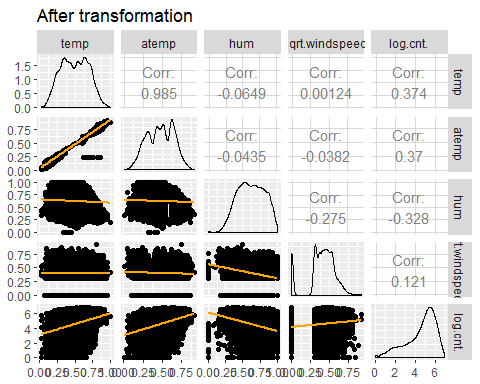
Let us perform initial univariate and bivariate analysis on the numerical dependent variables to get an idea of how they are distributed and how they vary when taken against the number of bikes rented **cnt**.

fn <- function(data,mapping)  
{  
 p = ggplot(data=data,mapping=mapping)+geom\_point()+geom\_smooth(method="lm",fill="orange",color="orange")  
 p  
}  
ggpairs(train[,c("temp","atemp","hum","windspeed","cnt")],lower=list(continuous=fn))+labs(title="Before transformation")



We see that there is significant skew in the density plots of **cnt** and windspeed. We perform a square root transformation on **windspeed** and a log transformation on **cnt**. We see that the linear relationship between the transformed count and the other numerical variables had become stronger. We also see that the transformed variables have become less skewed and help in ourlinear model as we will see later.

ggpairs(data.frame("temp"=train$temp,"atemp"=train$atemp,"hum"=train$hum,"sqrt(windspeed)"=sqrt(train$windspeed),"log(cnt)"=log(train$cnt)),lower=list(continuous=fn))+labs(title="After transformation")

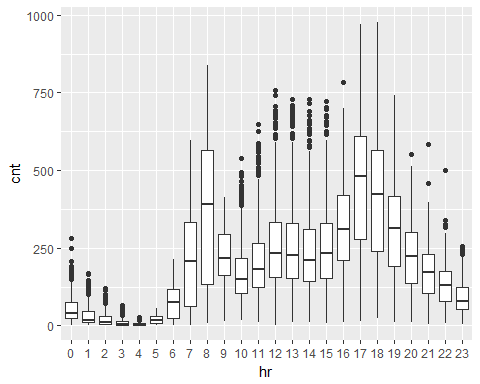


**Bivariate Analysis of count by hour of day**

The boxplots below convey the information that we would normally expect by common sense. We see that bicycle rentals are very low during the dead hours of the night. The rentals are quite high around 7-8 AM in the morning and 5-6 PM in the evening, they are relatively low during the afternoon hours and steadily decrease as the day ends

ggplot(train,aes(x=hr,y=cnt))+geom\_boxplot()

bike.day = aggregate(cnt~dteday,FUN=sum,data=bike)  
bike.day$dteday = as.Date(bike.day$dteday)  
bike.day$std.cnt = (bike.day$cnt-mean(bike.day$cnt))/sd(bike.day$cnt)



# Trivariate Analysis of categorical variables

We plot graphs to observe how **cnt** changes averaged by **hr** for the various levels of each categorical variables. We observe that for some of the categorical variables there are significant changes in the plots within the levels of that variable. This tells us that there is some interaction between the hour of the day and the other variables like the season and the day of the week.

g1=ggplot(train,aes(x=hr,y=cnt,colour=season,group=season))+stat\_summary(fun.y=mean,geom="point")+stat\_summary(fun.y=mean,geom="line")+labs(title="season)

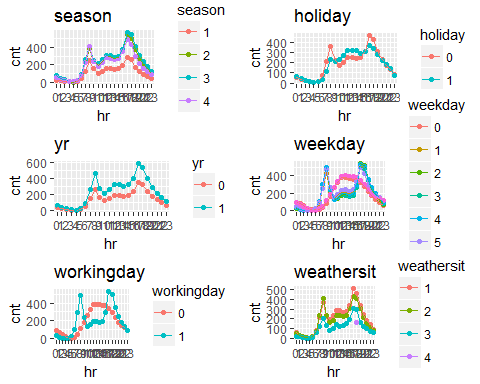
g2=ggplot(train,aes(x=hr,y=cnt,colour=holiday,group=holiday))+stat\_summary(fun.y=mean,geom="point")+stat\_summary(fun.y=mean,geom="line")+labs(title="holiday")

g3=ggplot(train,aes(x=hr,y=cnt,colour=yr,group=yr))+stat\_summary(fun.y=mean,geom="point")+stat\_summary(fun.y=mean,geom="line")+labs(title="yr")

g4=ggplot(train,aes(x=hr,y=cnt,colour=weekday,group=weekday))+stat\_summary(fun.y=mean,geom="point")+stat\_summary(fun.y=mean,geom="line")+labs(title="weekday")

g5=ggplot(train,aes(x=hr,y=cnt,colour=workingday,group=workingday))+stat\_summary(fun.y=mean,geom="point")+stat\_summary(fun.y=mean,geom="line")+labs(title="workingday")

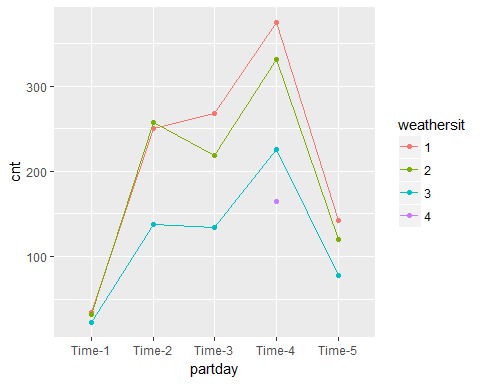
g6=ggplot(train,aes(x=hr,y=cnt,colour=weathersit,group=weathersit))+stat\_summary(fun.y=mean,geom="point")+stat\_summary(fun.y=mean,geom="line")+labs(title="weathersit")  
  
#graph 3  
grid.newpage()  
pushViewport(viewport(layout=grid.layout(3,2)))  
vplayout = function(x,y) viewport(layout.pos.row=x,layout.pos.col=y)  
print(g1,vp=vplayout(1,1))  
print(g2,vp=vplayout(1,2))  
print(g3,vp=vplayout(2,1))  
print(g4,vp=vplayout(2,2))  
print(g5,vp=vplayout(3,1))  
print(g6,vp=vplayout(3,2))



# Multivariate Analysis

Here we consider the variation of the response variable **cnt** taken again two or more variables at a time. We show only the most important interaction plots here given that we have a time, energy and space constraint. These plots help us determine if it is worthwhile adding interaction terms to our linear regression model. We divide the interactions into two types, one containing interactions amongst the weather variables like **temp** and **hum** and the other containing interactions relating to the calendar like "workingday" and **weekday**. We use some intuition and common sense here in choosing the plots though we use a more rigorous method to choose interaction terms for the model. Interaction for weather terms **Weather and Hour** We see that not all parts of the graph are parallel, meaning there is some interaction between the two variables. It also happens that many parts of the plots are parallel to each other meaning that the interaction is not very strong.

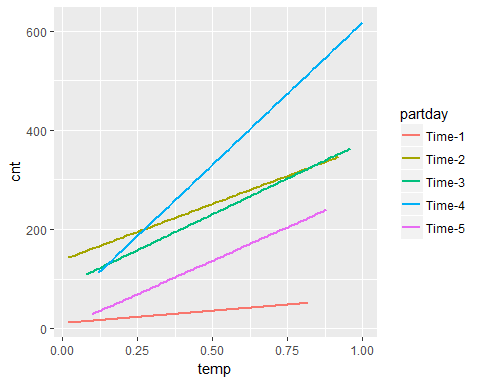
train$hum.bin = cut\_interval(train$hum,8)  
ggplot(train,aes(x=partday,y=cnt,colour=weathersit,group=weathersit))+stat\_summary(fun.y=mean,geom="point")+stat\_summary(fun.y=mean,geom="line")



**Temperature and Hour**

We can clearly see that there are differing slopes across the lines for different **partday**, though not all. On the whole, if any interaction is present, it would be very weak.

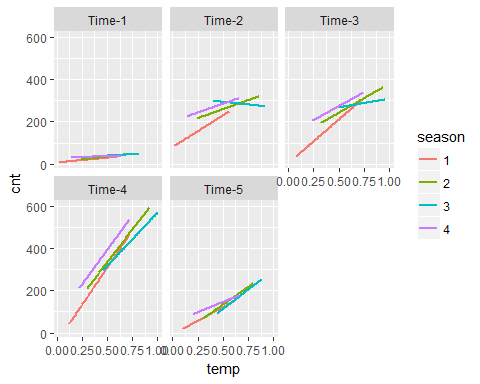
ggplot(train,aes(x=temp,y=cnt,group=partday,color=partday))+geom\_smooth(method="lm",se=FALSE)



**Temperature, Season and Hour**

We take the various combination pairs of **partday** and **season** and construct fitted line plots with temperature. We fix one categorical variable at one level and see if it varies while varying the other variable. We can see that there is a lot of variability in the slopes of the different lines. We will consider adding this interaction term.

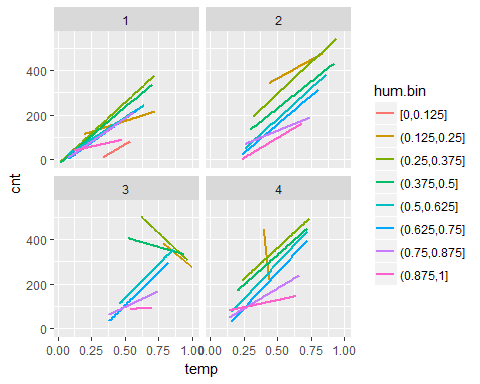
ggplot(train,aes(x=temp,y=cnt,group=season,color=season))+geom\_smooth(method="lm",se=FALSE)+facet\_wrap(~partday,drop=FALSE)



**Temperature and Humidity and Season**

Similar to the above case, we find a lot of variation in slopes. There must be some interaction present and we will consider it in our model.

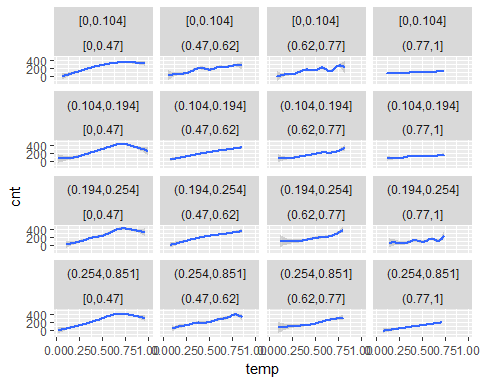
ggplot(train,aes(x=temp,y=cnt,group=hum.bin,color=hum.bin))+geom\_smooth(method="lm",se=FALSE)+facet\_wrap(~season,drop=FALSE)



**Temperature and Humidity and Windspeed**

By observing the variations in the fitted loess curves in each sub-plot, we can say that there is not enough proof that there exists a three-way interaction between the above variables. We show later that the interaction is not significant.

ggplot(train,aes(x=temp,y=cnt))+geom\_smooth()+facet\_wrap(~cut\_number(windspeed,4)+cut\_number(hum,4),drop=FALSE)



# Multiple Linear Regression model with interactions

We build the model allowing for a maximum of three-way interactions with the transformed variables and a little bit of intuition and common sense. We then use the Type II ANOVA tests to trim the regressor combination to size by eliminating insignificant interactions. We obtain a very good fit with a good value.

#Full model with interactions  
dayint = "yr+weekday+workingday+weekday:hr+workingday:hr"  
form.log = paste0("log(cnt)~",weatherint,"+",dayint)  
form = formula(paste0("cnt~",weatherint,"+",dayint))  
m.log = lm(formula=form.log,data=train)  
m = lm(formula=form,data=train)

# Dispaly Anova table

Anova(m.log)

## Anova Table (Type II tests)  
## Response: log(cnt)  
## Sum Sq Df F value Pr(>F)   
## temp 244.4 1 2031.1612 < 2.2e-16 \*\*\*  
## hum 12.4 1 103.2157 < 2.2e-16 \*\*\*  
## sqrt(windspeed) 8.5 1 70.7859 < 2.2e-16 \*\*\*  
## weathersit 162.2 3 449.4366 < 2.2e-16 \*\*\*  
## season 409.9 3 1135.5359 < 2.2e-16 \*\*\*  
## hr 12687.7 23 4584.8263 < 2.2e-16 \*\*\*  
## yr 583.5 1 4850.0333 < 2.2e-16 \*\*\*  
## weekday 39.9 6 55.2357 < 2.2e-16 \*\*\*  
## workingday 0.8 1 6.4128 0.0113447 \*   
## temp:hum 6.9 1 57.6211 3.451e-14 \*\*\*  
## temp:sqrt(windspeed) 0.7 1 5.9413 0.0148064 \*   
## temp:season 61.1 3 169.2991 < 2.2e-16 \*\*\*  
## hum:season 15.9 3 44.0949 < 2.2e-16 \*\*\*  
## sqrt(windspeed):season 2.3 3 6.4325 0.0002392 \*\*\*  
## season:hr 19.8 69 2.3862 1.086e-09 \*\*\*  
## temp:hr 12.7 23 4.5763 2.041e-12 \*\*\*  
## hr:weekday 169.9 138 10.2328 < 2.2e-16 \*\*\*  
## hr:workingday 110.4 23 39.8995 < 2.2e-16 \*\*\*  
## temp:hum:season 16.8 3 46.5314 < 2.2e-16 \*\*\*  
## temp:sqrt(windspeed):season 1.0 3 2.6579 0.0466151 \*   
## temp:hum:hr 3.5 23 1.2805 0.1661509   
## hum:sqrt(windspeed):hr 5.0 24 1.7401 0.0138601 \*   
## hum:sqrt(windspeed):season 0.4 3 1.0531 0.3677672   
## temp:season:hr 12.7 69 1.5352 0.0029491 \*\*   
## hum:season:hr 11.1 69 1.3314 0.0349052 \*   
## Residuals 1249.7 10387   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Testing the fit using RMSLE**

We use the Root Mean Square Log Error to check how well our model predicts.

test = bike[bike$type=="test",-18]  
test$p.log.cnt = predict(m.log,newdata=test)  
paste0("RMSLE: ",rmsle(test$cnt,exp(test$p.log.cnt)))

## [1] "RMSLE: 0.384828718281902"

write.csv(exp(test$p.log.cnt),file="out.csv")

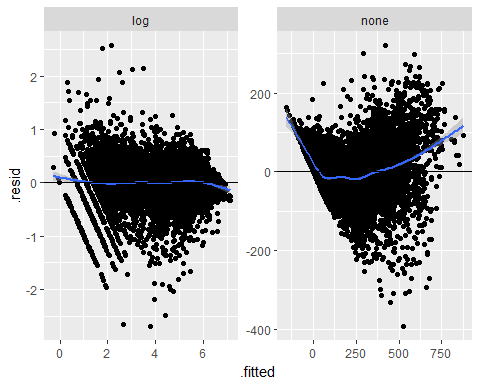
**Graphical evaluation of the model**

We plot the residual vs fitted plot to check for any violations of the linear regression assumptions. We draw two such plots one where we have used the log transformation on "cnt" and the other without it. We see that the loess curve in the log case almost completely coincides with the the zero line and the confidence interval engulfs the zero line almost. The non-linear curve in the non-log case reflects model inadequacies, and the transformation significantly improves the plot

lm.m = augment(m)

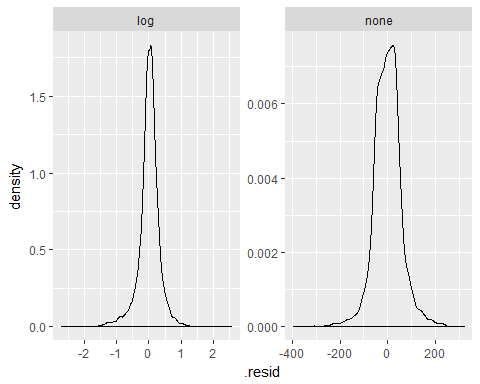
lm.m.log = augment(m.log)

names(lm.m.log)[1] = "cnt"  
lm.m$transf = "none"  
lm.m.log$transf = "log"  
lm.df = rbind(lm.m,lm.m.log)  
lm.df$transf = as.factor(lm.df$transf)  
  
ggplot(data=lm.df,aes(x=.fitted,y=.resid))+geom\_point()+geom\_smooth()+geom\_abline(slope=0,intercept=0)+facet\_wrap(~transf,scales="free")



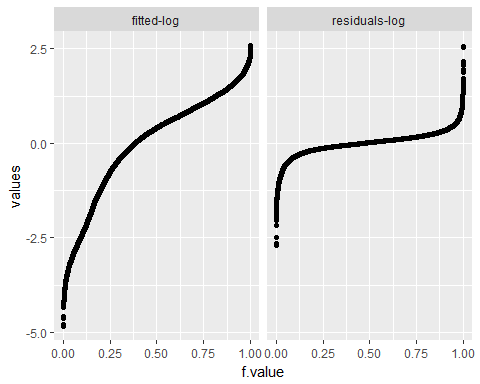
We check for normality of residuals and we see that the residuals when pooled follow a normal distribution clearly seen from the bell shape of the kernel density curve. We see a slight improvement after the log transformation.

ggplot(lm.df,aes(x=.resid))+geom\_density()+facet\_wrap(~transf,scales="free")



We plot the fitted values and residuals against a common scale to check if the model does well enough in explaining the variation in "cnt". We see that fitted values are more spread out than the residuals. This means that our model performs well enough.

n = nrow(lm.df)/2  
f.value = (0.5:(n-0.5))/n  
fitted = sort(lm.m.log$.fitted)-mean(lm.m.log$.fitted)  
residual = sort(lm.m.log$.resid)  
df.1 = data.frame("values"=c(fitted,residual),"type"=c(rep("fitted-log",n),rep("residuals-log",n)),"f.value"=c(f.value,f.value))  
ggplot(df.1,aes(x=f.value,y=values))+geom\_point()+facet\_wrap(~type)



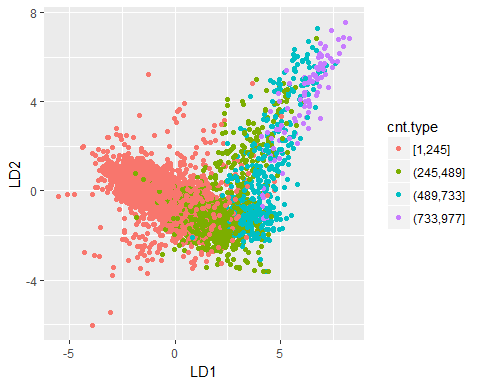
**Major limitations of the linear model**

Though our linear model is able to capture a lot of the variation, it is highly dependent on higher order interactions. We have to be careful in choosing the interactions as choosing too many of the higher order interactions blindly might lead to multicollinearity.By including too many interactions we also run the risk of over fitting.

# Linear Discriminant Analysis

We bucket the variable **cnt** into four different range buckets and perform LDA to see if we achieve separation between the classes. We DO NOT intend to use the model to predict. Our aim is purely an investigative one.

bike$cntbin = cut\_interval(bike$cnt,4)  
train = bike[bike$type=="train",-18]  
test = bike[bike$type=="test",-18]  
cntbin = test$cntbin  
lda.form = formula("cntbin~temp+hum+sqrt(windspeed)+weathersit+season+hr+temp:hum+temp:sqrt(windspeed)+temp:season+hum:season+sqrt(windspeed):season+temp:hum:season+temp:sqrt(windspeed):season+temp:hum:hr+sqrt(windspeed):hum:hr+hum:sqrt(windspeed):season+hr:season+temp:hr:season+hum:hr:season+yr+weekday+workingday+weekday:hr+workingday:hr")  
m.lda = lda(lda.form,data=train)  
p.lda = predict(m.lda,newdata=test)  
ggplot(data.frame(LD1=p.lda$x[,1],LD2=p.lda$x[,2],cnt.type=cntbin),aes(x=LD1,y=LD2,group=cnt.type,color=cnt.type))+geom\_point()



We see that we are able to achieve a decent level of separation from the LDA components plot. We also present the accuracy of the model just to quench your curiosity.

sum(diag(table(actual=test$cntbin,pred=p.lda$class)))/sum(table(actual=test$cntbin,pred=p.lda$class))

## [1] 0.8469121

**Time components in the data**

We shall try to extract the time components from the data to see to what extent a time series analysis is possible. We see from the extracted components below that it is an increasing time series and it is not difficult to figure out that there exists monthly seasonality. However the residual (noise) component of the variation is quite large and looks like it may not be white noise.

tm = lm(std.cnt~dteday,data=bike.day)  
tm.df = augment(tm)

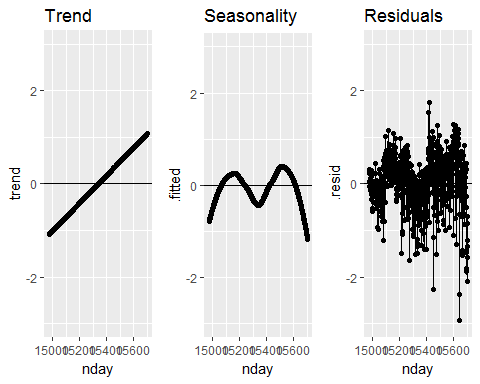
trend = tm.df$.fitted-mean(tm.df$.fitted)  
nday = as.numeric(bike.day$dteday)  
g1=ggplot(bike.day,aes(x=nday,y=std.cnt))+geom\_point()+geom\_line()+geom\_smooth()+labs(title="Plot by day")

g2=ggplot(tm.df,aes(x=nday,y=trend))+geom\_point()+geom\_line()+geom\_abline(slope=0,intercept=0)+labs(title="Trend")+ylim(-3,3)  
tlm = loess(tm.df$.resid~nday)  
tlm.df = augment(tlm)

g3=ggplot(tlm.df,aes(x=nday,y=.fitted))+geom\_point()+geom\_line()+geom\_abline(slope=0,intercept=0)+labs(title="Seasonality")+ylim(-3,3)

g4=ggplot(tlm.df,aes(x=nday,y=.resid))+geom\_point()+geom\_line()+geom\_abline(slope=0,intercept=0)+labs(title="Residuals")+ylim(-3,3)

grid.newpage()  
pushViewport(viewport(layout=grid.layout(1,3)))  
vplayout = function(x,y) viewport(layout.pos.row=x,layout.pos.col=y)  
print(g2,vp=vplayout(1,1))  
print(g3,vp=vplayout(1,2))  
print(g4,vp=vplayout(1,3))



# Future work

We could consider using an ARIMA model to see if time series analysis provides better prediction results. Also, since we are dealing with the count of the rented bikes we could consider fitting a Poisson regression model ,Tree based ML algorithms and Anomaly detection for getting some more better prediction.