Appendix A – EDA and Linear Regression

Nick Wawee

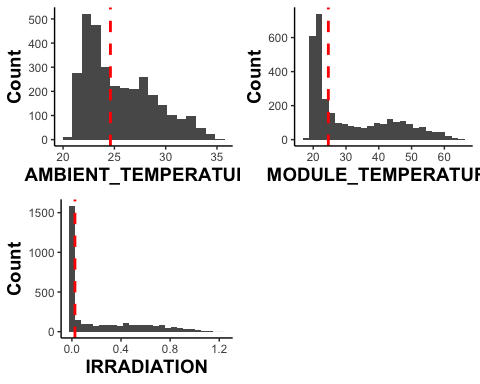
7/3/2021

## Loading - Power Plant 1

df1 = read.csv('Data/Plant\_1\_Generation\_Data.csv', stringsAsFactors = T)  
df1$DATE\_TIME = strptime(as.character(df1$DATE\_TIME), format = "%d-%m-%Y %H:%M")#Converting to timestamp  
  
wdf1 = read.csv('Data/Plant\_1\_Weather\_Sensor\_Data.csv', stringsAsFactors = T)  
wdf1$DATE\_TIME = strptime(as.character(wdf1$DATE\_TIME), format = "%Y-%m-%d %H:%M:%S")#Conerting to timestamp  
  
#Joining Datasets  
wdf1 = wdf1[,c('DATE\_TIME', "AMBIENT\_TEMPERATURE", "MODULE\_TEMPERATURE", "IRRADIATION" )]  
df = merge(df1, wdf1)

### Distribution of Weather Variables

#plotdists(df1[,c(-1,-2,-3)],'Plots/p1\_generation\_', brtype = 'Scott')  
plotdists(wdf1[,-1],'Plots/p1\_weather\_',)

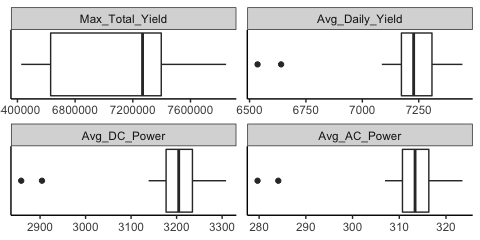


### Inverter EDA

#Max Yields  
maxyields = data.frame('SOURCE\_KEY' = as.character(), 'Max\_Total\_Yield' = as.numeric())  
for (key in levels(df$SOURCE\_KEY)){  
 yields = df$TOTAL\_YIELD[df$SOURCE\_KEY==key]  
 dfa = data.frame('SOURCE\_KEY' = key, 'Max\_Total\_Yield' = max(yields))  
 maxyields = rbind(dfa, maxyields)  
}  
  
#Average Daily Yield  
dailyyield = data.frame('SOURCE\_KEY' = as.character(), 'Avg\_Daily\_Yield' = as.numeric(),'Avg\_DC\_Power' = as.numeric(), 'Avg\_AC\_Power' = as.numeric())  
  
for (key in levels(df$SOURCE\_KEY)){  
 keydf = df[df$SOURCE\_KEY==key,]  
 keydf$dates = factor(as.Date(keydf$DATE\_TIME))  
 maxvec = as.numeric()  
 dcvec = as.numeric()  
 acvec = as.numeric()  
 for (d in levels(keydf$dates)){  
 datedf = keydf[keydf$dates == d,]  
 maxvec = c(max(datedf$DAILY\_YIELD), maxvec)  
 dcvec = c(mean(datedf$DC\_POWER), dcvec)  
 acvec = c(mean(datedf$AC\_POWER), acvec)  
 }  
 dfa = data.frame('SOURCE\_KEY' = key, 'Avg\_Daily\_Yield' = mean(maxvec), 'Avg\_DC\_Power' = mean(dcvec), 'Avg\_AC\_Power' = mean(acvec) )  
 dailyyield = rbind(dfa, dailyyield)  
}  
  
yielddf = merge(maxyields, dailyyield)  
  
yielddf.m = melt(yielddf)

## Using SOURCE\_KEY as id variables

ggplot(data = yielddf.m, aes(x = value))+geom\_boxplot()+facet\_wrap(~variable, scales = 'free')+plot\_opts+theme(axis.text.y = element\_blank(), axis.ticks.y = element\_blank(), axis.title.x = element\_blank())



ggsave('Plots/invertereda.pdf', dpi = 600)

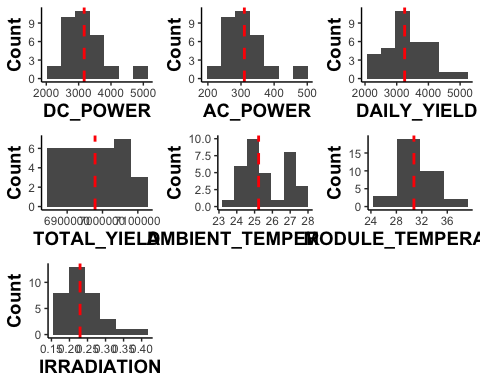
## Saving 5 x 2.5 in image

It looks like there are two outliers for each, the worse performing source key is 1BY6WEcLGh8j5v7 while the other is bvBOhCH3iADSZry .

### Time EDA

Similarly, daily values of all variables will be analyzed.

df$DATES = factor(as.Date(df$DATE\_TIME))  
dailydf = data.frame()  
  
for (d in levels(df$DATES)){  
 datedf = df[df$DATES==d,]  
 datedf = datedf[,c(-1,-2,-3,-length(datedf))]  
 dfnew = t(as.data.frame(colMeans(datedf)))  
 row.names(dfnew) = d  
 dailydf = rbind(dfnew, dailydf)  
}  
  
#dailydf$DATE = row.names(dailydf)  
  
plotdists(dailydf, path = 'Plots/dailyeda')



write.csv(x = df, file = 'Data/Plant\_1\_Clean.csv')

### ARIMA Model

Below will aggregate all ac power measurements for the plant.

dt = factor(df$DATE\_TIME)  
dfnew = data.frame('DATE\_TIME' = rep(NA,length(levels(dt))), 'Avg\_AC' = rep(NA,length(levels(dt))))  
  
i = 1  
for (d in levels(dt)){  
 dtdf = df[df$DATE\_TIME == d, ]  
 dfnew$DATE\_TIME[i] = d  
 dfnew$Avg\_AC[i] = sum(dtdf$DAILY\_YIELD)  
 i = i +1  
}  
dfnew$Sum\_AC = dfnew$Avg\_AC  
dfnew = dfnew[,-2]

write.csv(x = dfnew, file = 'Data/aggregated\_yield.csv')

## Linear Regression

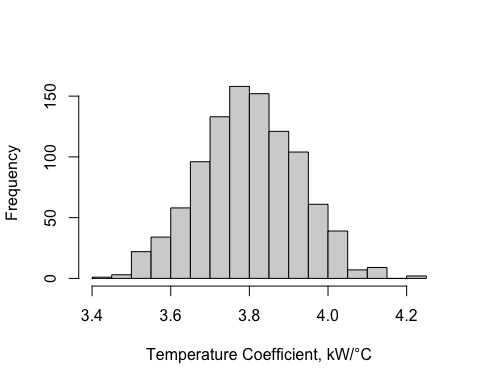
regdf = df[,c('DC\_POWER', 'IRRADIATION', 'MODULE\_TEMPERATURE')]  
lmodel = lm(DC\_POWER~0 +MODULE\_TEMPERATURE + IRRADIATION, data=regdf)  
summary(lmodel)

##   
## Call:  
## lm(formula = DC\_POWER ~ 0 + MODULE\_TEMPERATURE + IRRADIATION,   
## data = regdf)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16183.1 -85.7 -77.8 96.7 7349.6   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## MODULE\_TEMPERATURE 3.796e+00 1.249e-01 30.39 <2e-16 \*\*\*  
## IRRADIATION 1.306e+04 1.101e+01 1186.19 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 585.5 on 68772 degrees of freedom  
## Multiple R-squared: 0.9869, Adjusted R-squared: 0.9869   
## F-statistic: 2.594e+06 on 2 and 68772 DF, p-value: < 2.2e-16

rmse = sqrt(mean(lmodel$residuals\*\*2))  
rmse

## [1] 585.4585

n.sims = 1000  
sim.1 <- sim (lmodel, n.sims)  
  
temp.coef <- sim.1@coef[,1]  
irr.coef =sim.1@coef[,2]  
  
hist(temp.coef, xlab = 'Temperature Coefficient, kW/°C', main = '')



hist(irr.coef, xlab = 'Irradiation Coefficient, m^2', main = '')

