Prediction of Bike Rental Count Per day

Bharat Mohan Thakur

22-12-2019

Contents

Chapter 1: Introduction

1.1 Problem Statement

The objective of this project is to predict the count of bike rentals day wise.. Project will help accommodate in managing the number of bikes required on a daily basis, and being prepared for high demand of bikes during peak periods.

1.2 Data

The objective is to build regression models which predicts the number of bikes. Given below is a sample of the data set t

Table 1.1: Bike Count Sample Data (Columns: 1-9)

instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit
1	2011-01-01	1	0	1	0	6	0	2
2	2011-01-02	1	0	1	0	0	0	2
3	2011-01-03	1	0	1	0	1	1	1
4	2011-01-04	1	0	1	0	2	1	1
5	2011-01-05	1	0	1	0	3	1	1

Table 1.2: Bike Count Sample Data (Columns: 10-16)

temp	atemp	hum	windspeed	casual	registered	cnt
0.3441670	0.3636250	0.805833	0.1604460	331	654	985
0.3634780	0.3537390	0.696087	0.2485390	131	670	801
0.1963640	0.1894050	0.437273	0.2483090	120	1229	1349
0.2000000	0.2121220	0.590435	0.1602960	108	1454	1562
0.2269570	0.2292700	0.436957	0.1869000	82	1518	1600

table below we have the following 13 variables, using which we have to correctly predict the count of bikes:

Column_no	Columkn_name
1	Instant
2	Dteday
3	Season
4	Yr
5	Month
6	Holiday
7	Weekday
8	Workingday
9	Weathersit
10	Temp
11	Atemp
12	Hum
13	windspeed

Table 1.3: Predictor variables

Chapter 2: Exploratory Data Analysis

2.1 Pre-Processing

We should look at the data before we start to create a model. In data mining, looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is known as Exploratory Data Analysis.

2.2 Scatter plot of all variables against each other

It can be observed from the below scatter plots that temp and atemp has high correlation with each other and humidity ,windspeed and cnt is normally distributed .

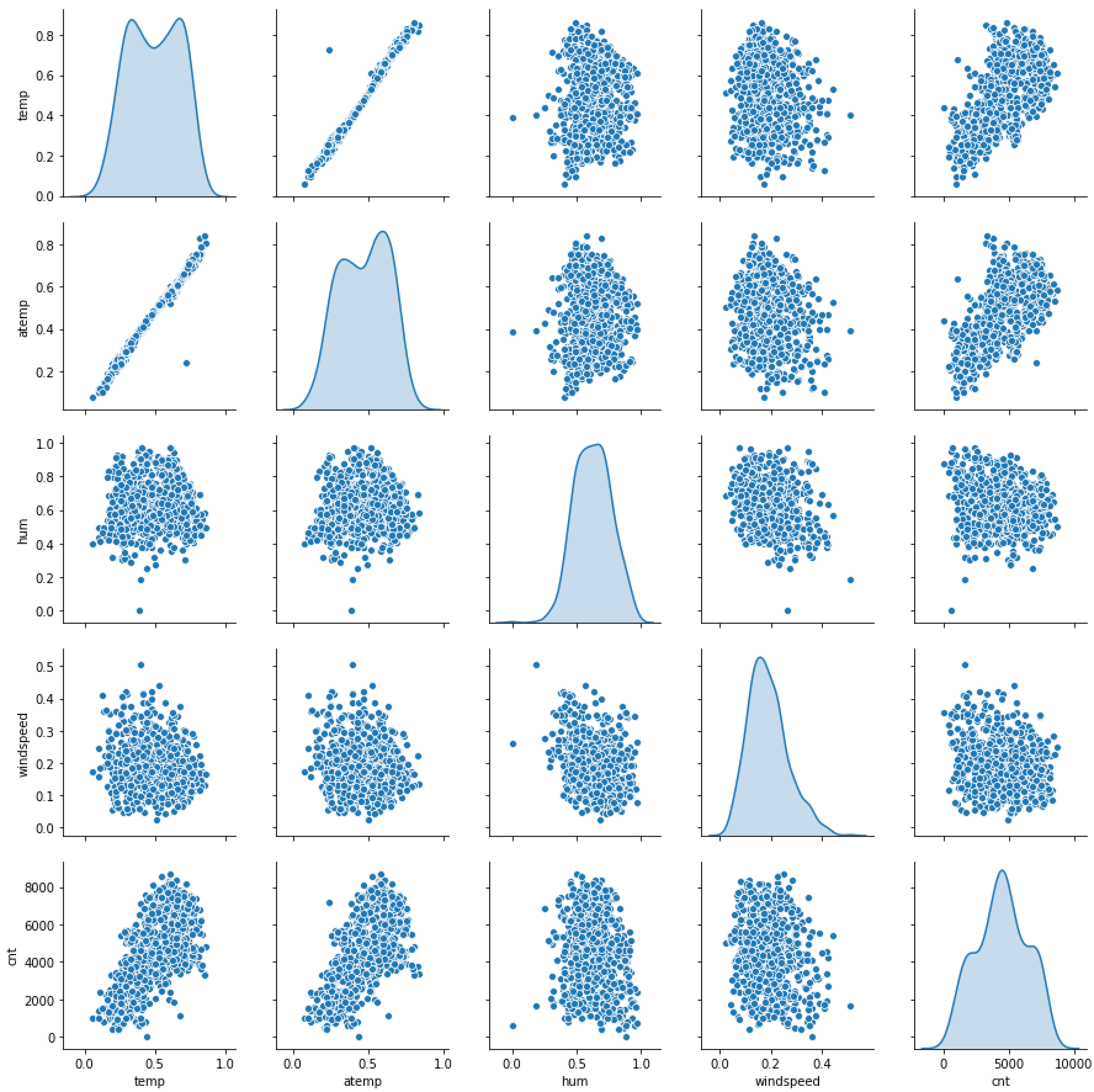
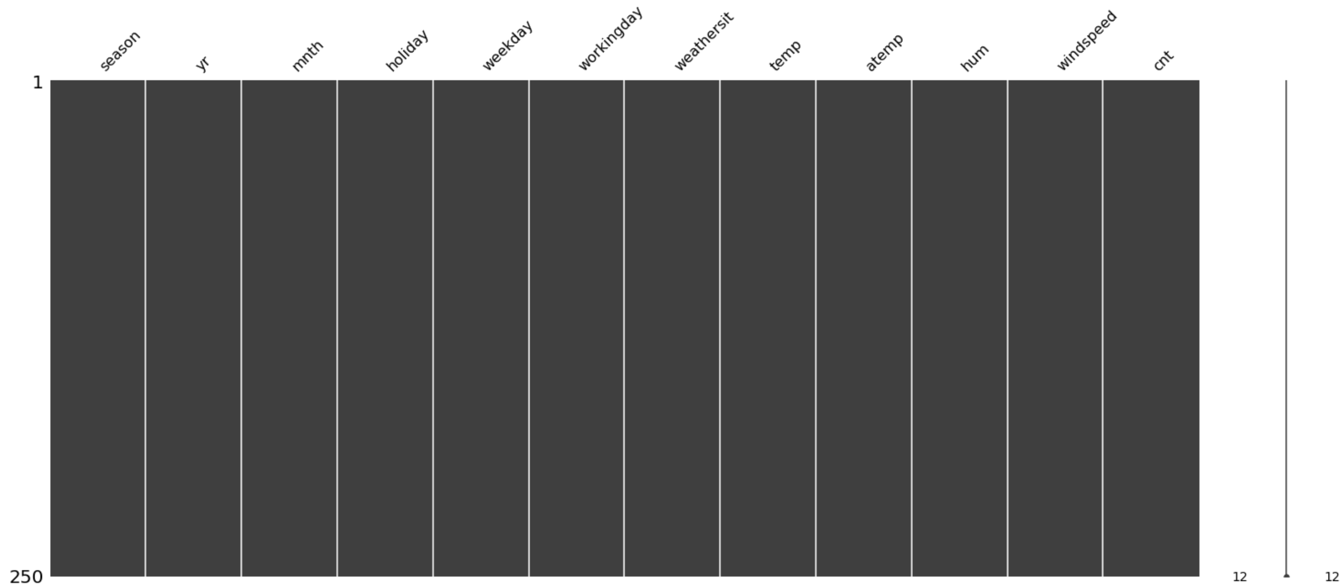
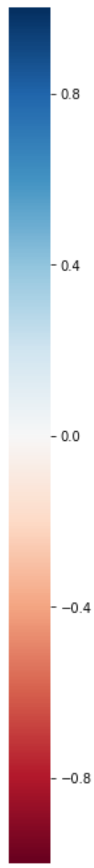
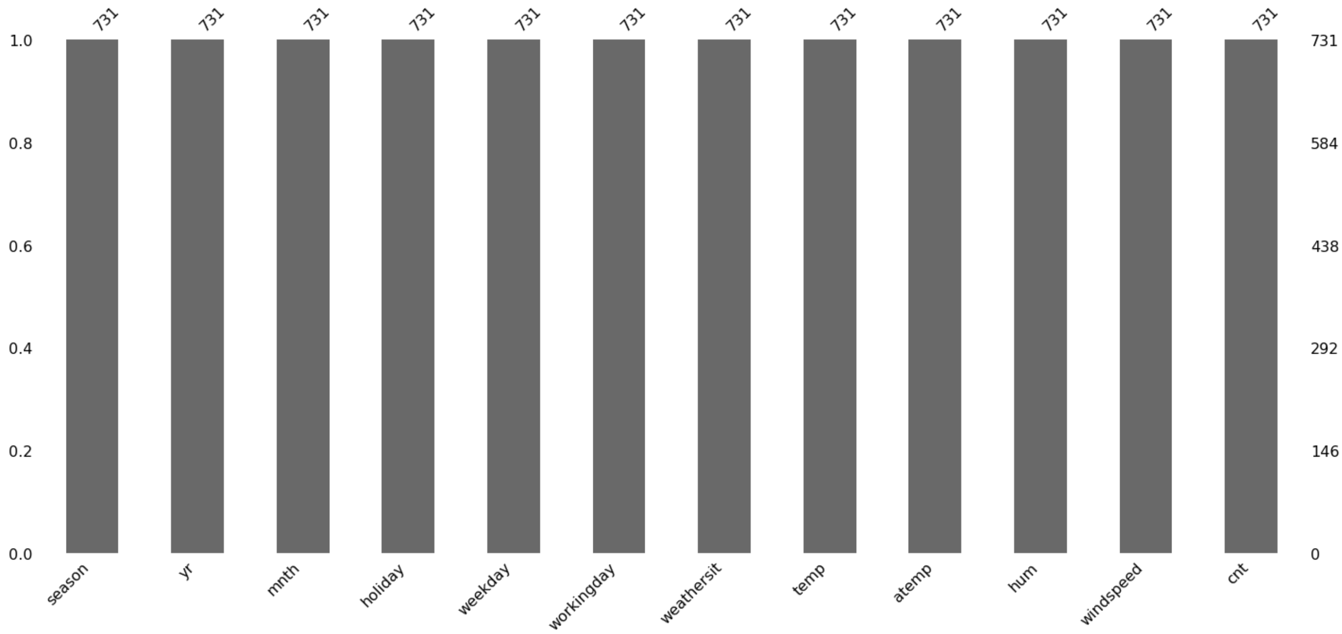


Fig 2.1: Scatter plot of all numerical variables

2.3 missing value analysis (using missigno library)



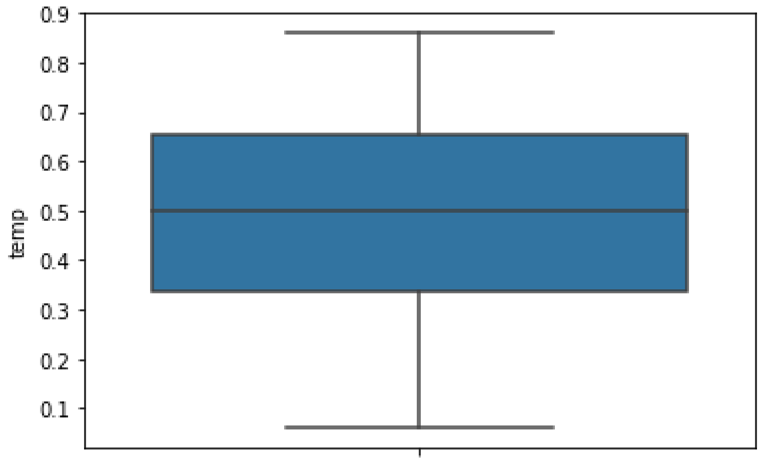




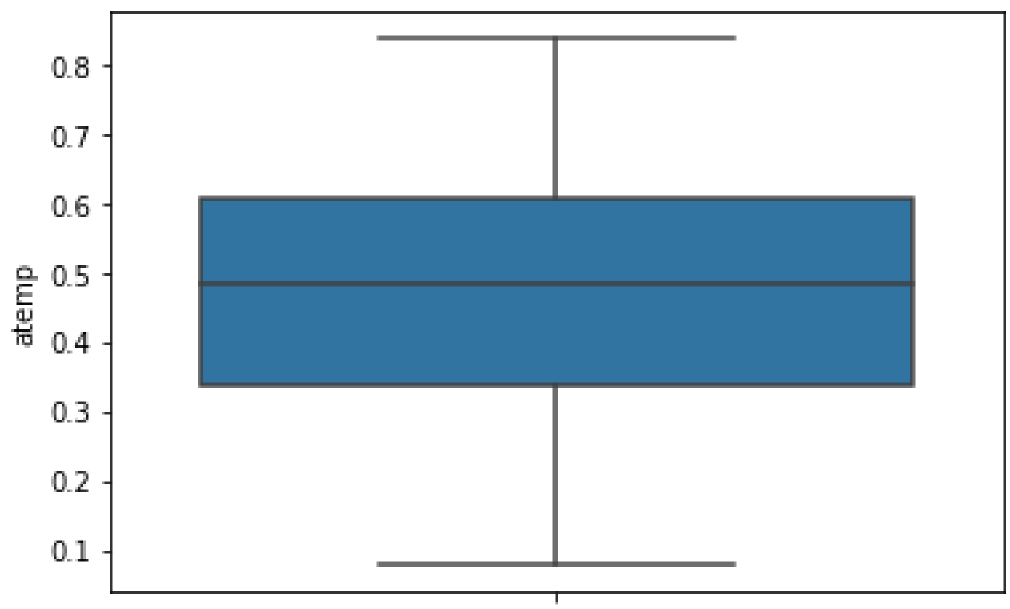
Barplot of missing value shows that there is no missing value in any row

2.3 outlier analysis (creating boxplot)

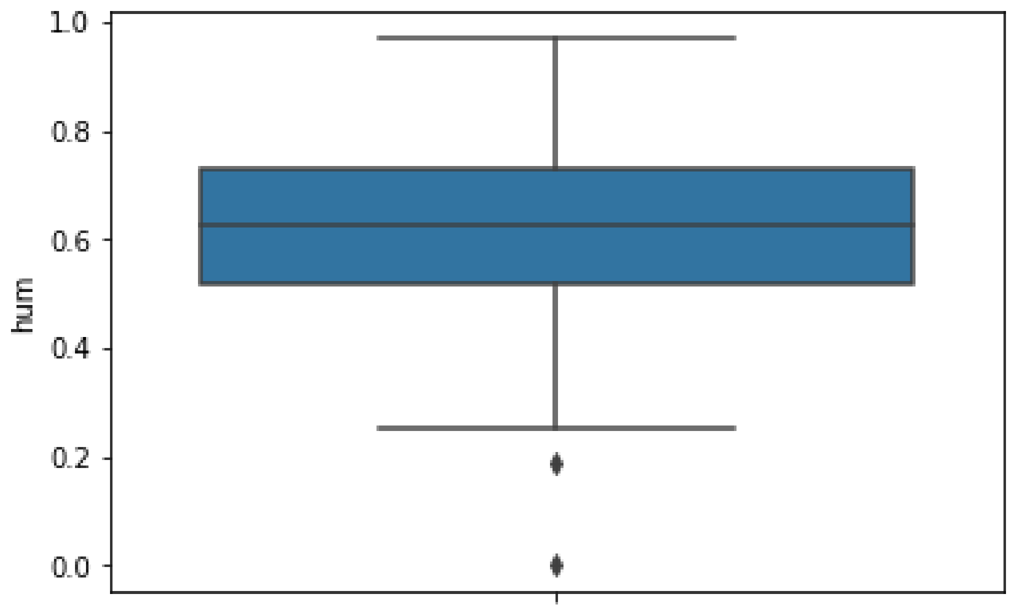
2.3.1 Boxplot of temp show that there is no outlier .



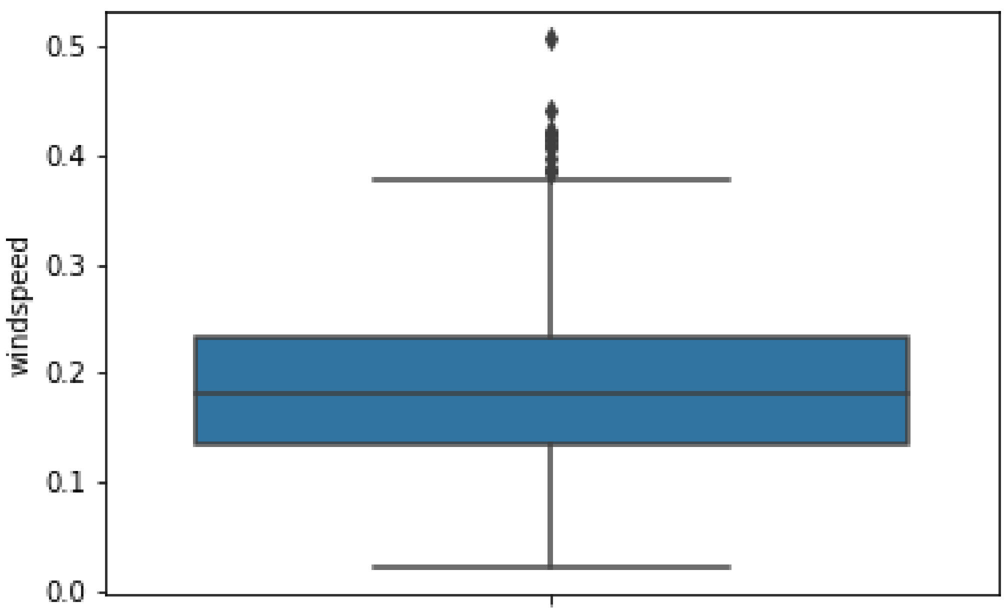
2.3.2 Boxplot of atemp show that there is no outlier .



2.3.2 Boxplot of hum show that there is outlier and need to be treated .



2.3.2 Boxplot of windspeed show that there is outlier and need to be treated .



Conclusion : it is clear from boxplot that there is outlier in windspeed and hum has outlier

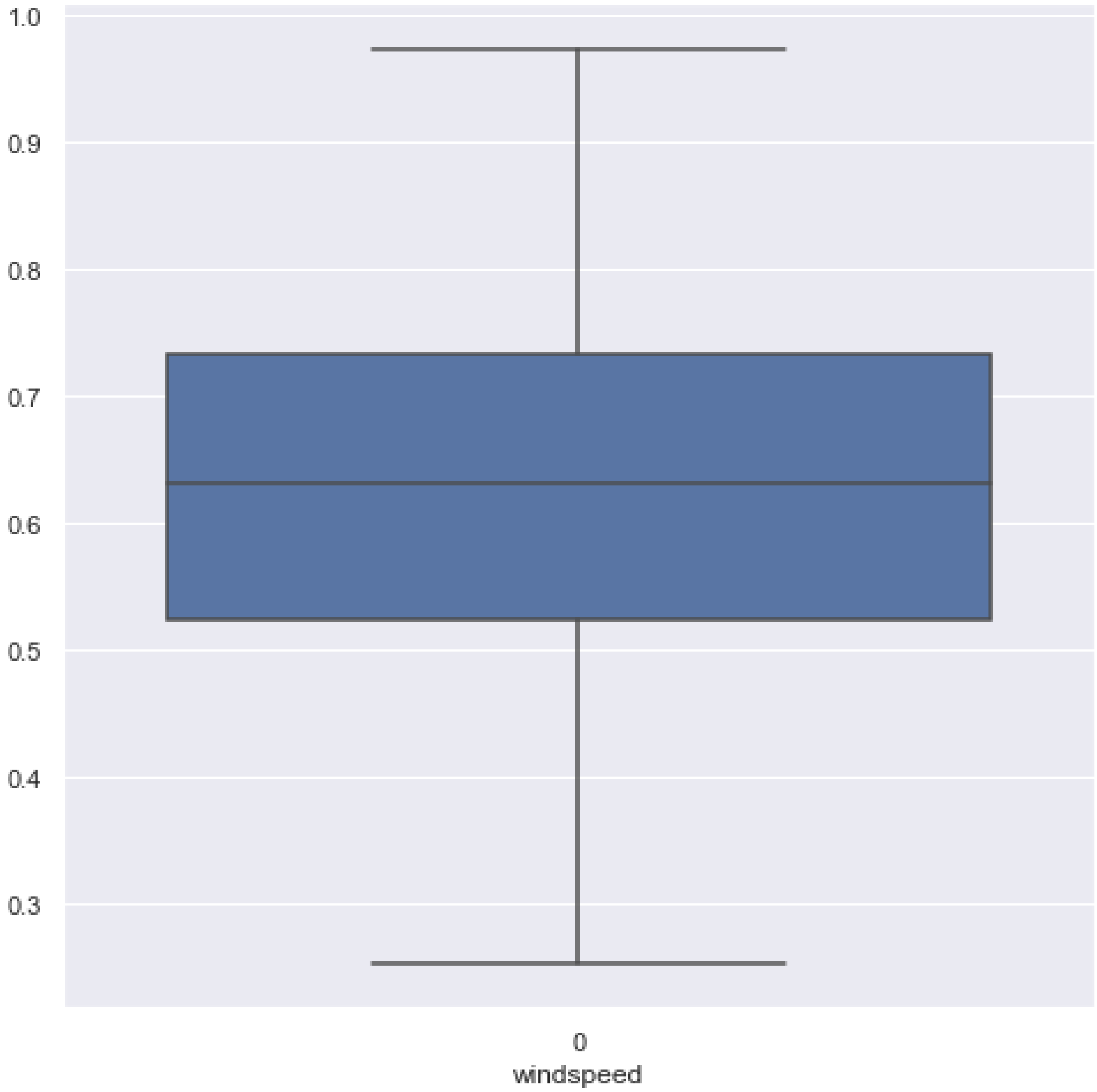
2.4 outlier removal by using interquartile range :

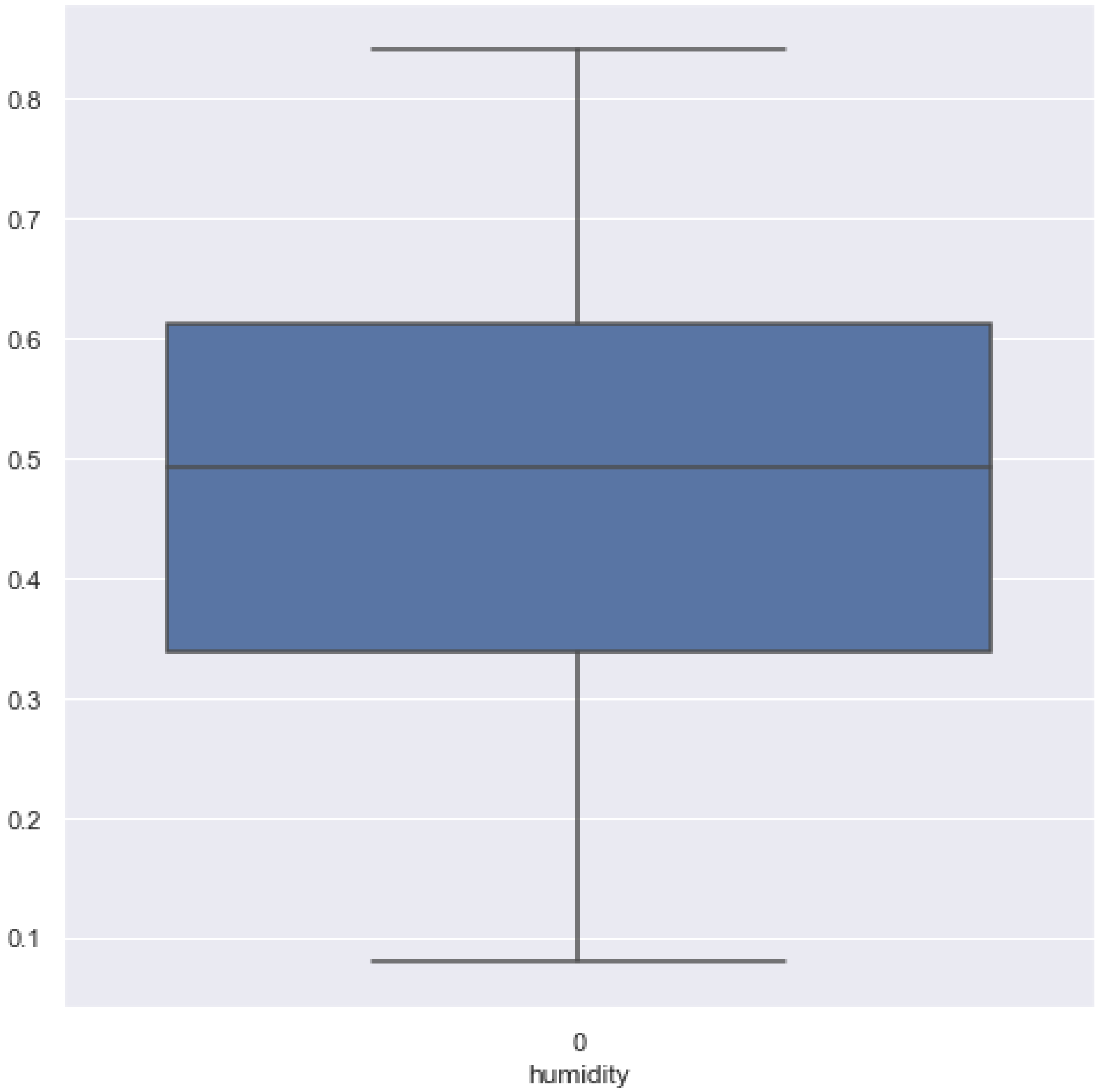
Code

day_bike = day_bike[~((day_bike[numerical_columns] < (Q1 - 1.5 * IQR)) |(day_bike[numerical_columns] > (Q3 + 1.5 * IQR))).any(axis=1)]

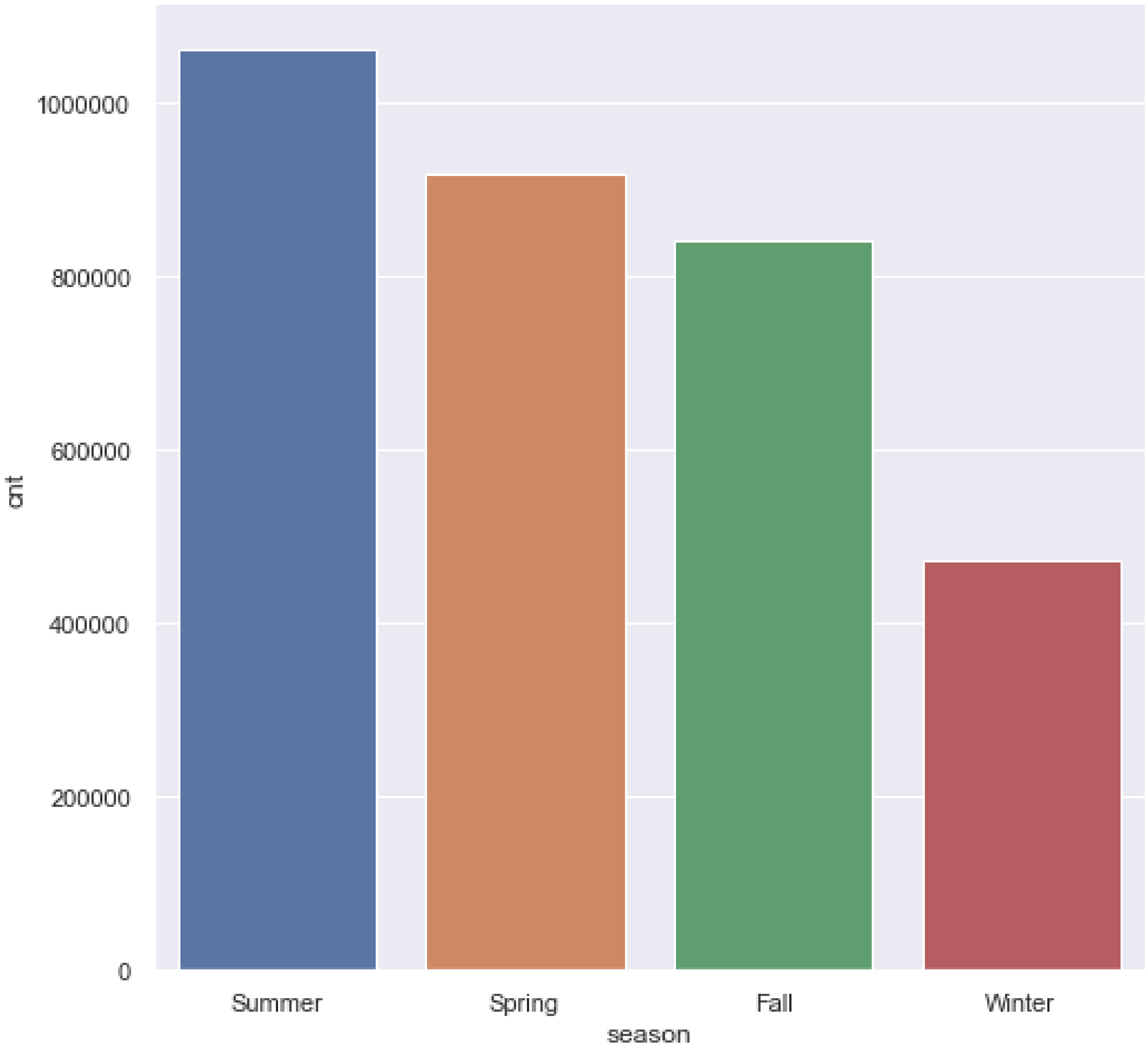
after outlier removal we are with 717 rows which means 14 rows are dropped

humidity after outlier removal


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
windspeed



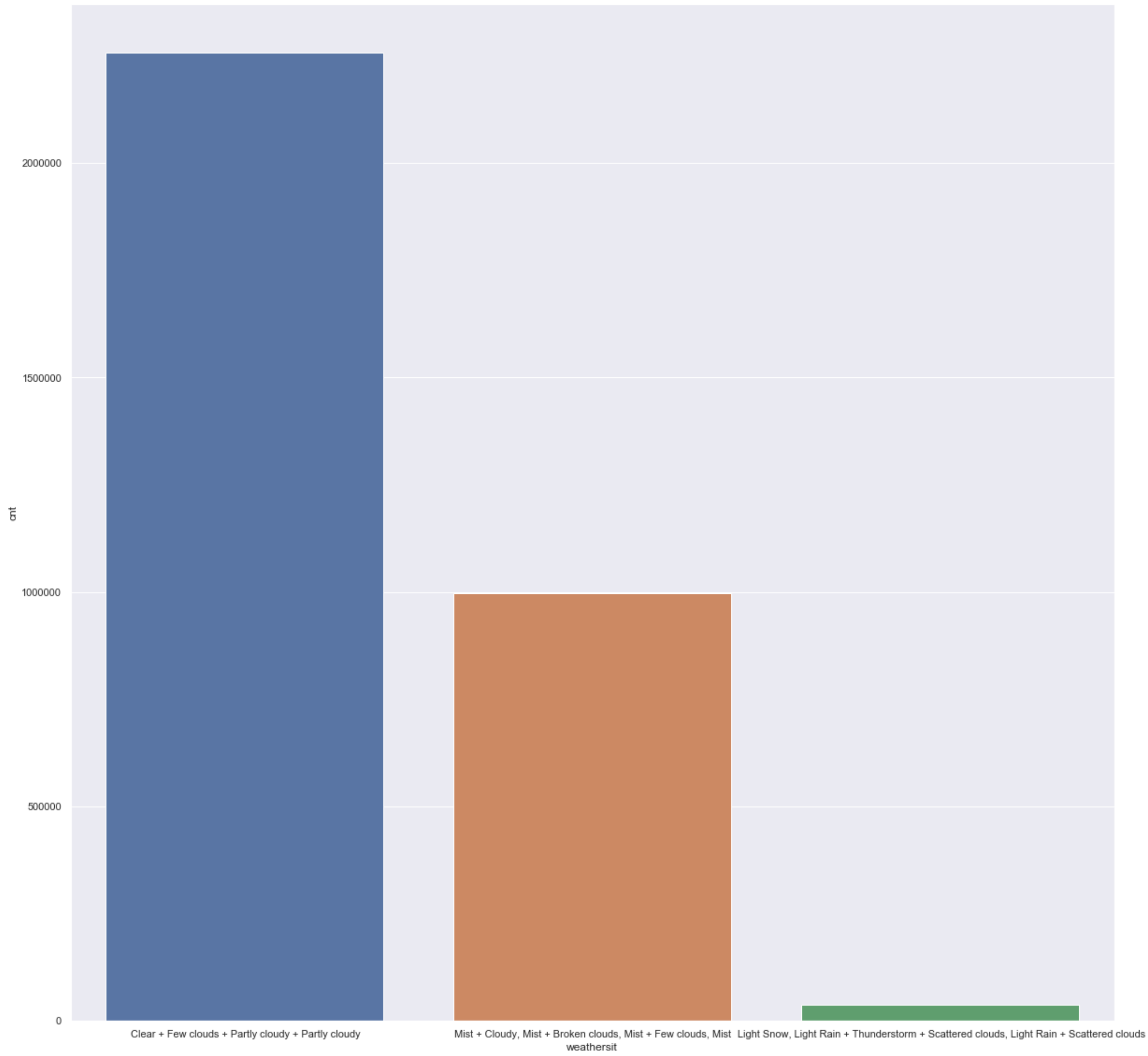
Chrecking distribution of categorical columns wrt cnt by using barplot:



it can be clearly inferred from above that summer the cnt of vehicle is maximum and winter it is least so we can conclude that season has effect on bike count

weathersit vs cnt :

it can be clearly inferred from below graph that as weather goes bad bike cnt decreases so there is direct realation.



Holiday vs cnt

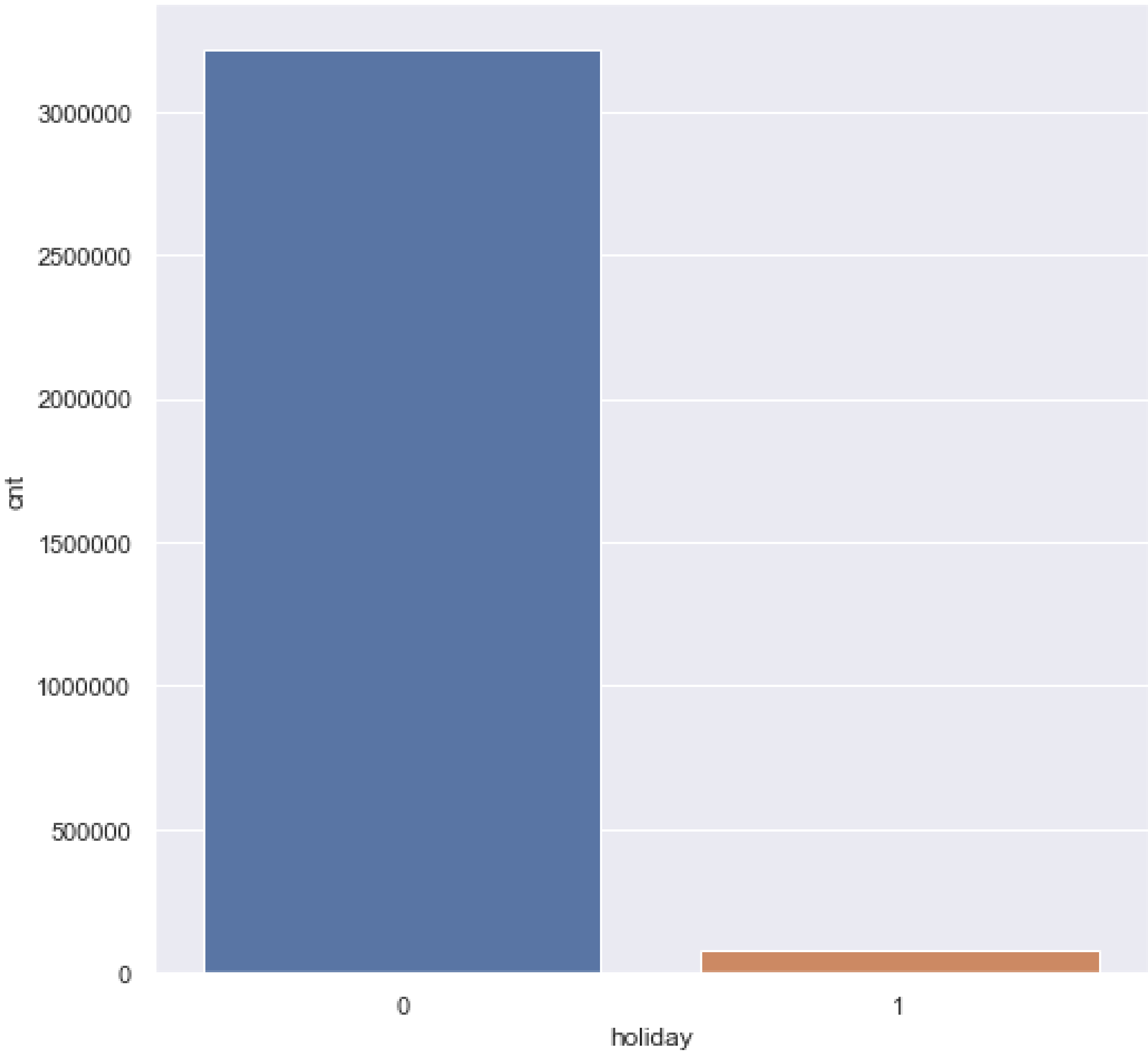
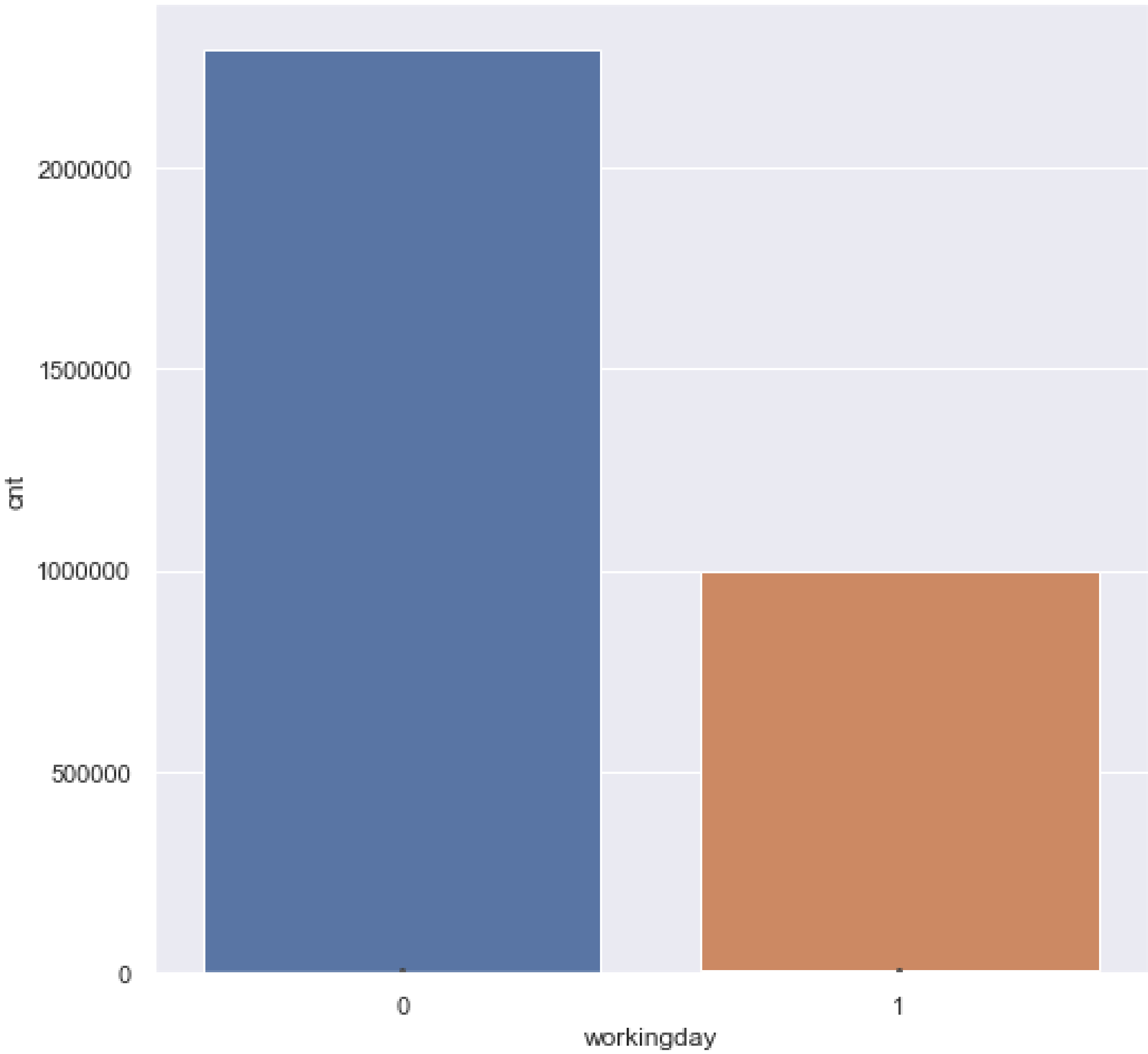


Fig 2.2: Distribution of categorical variables using bar plots

It can be seen from above barplot that in holiday maximum bike in working day it is negligible

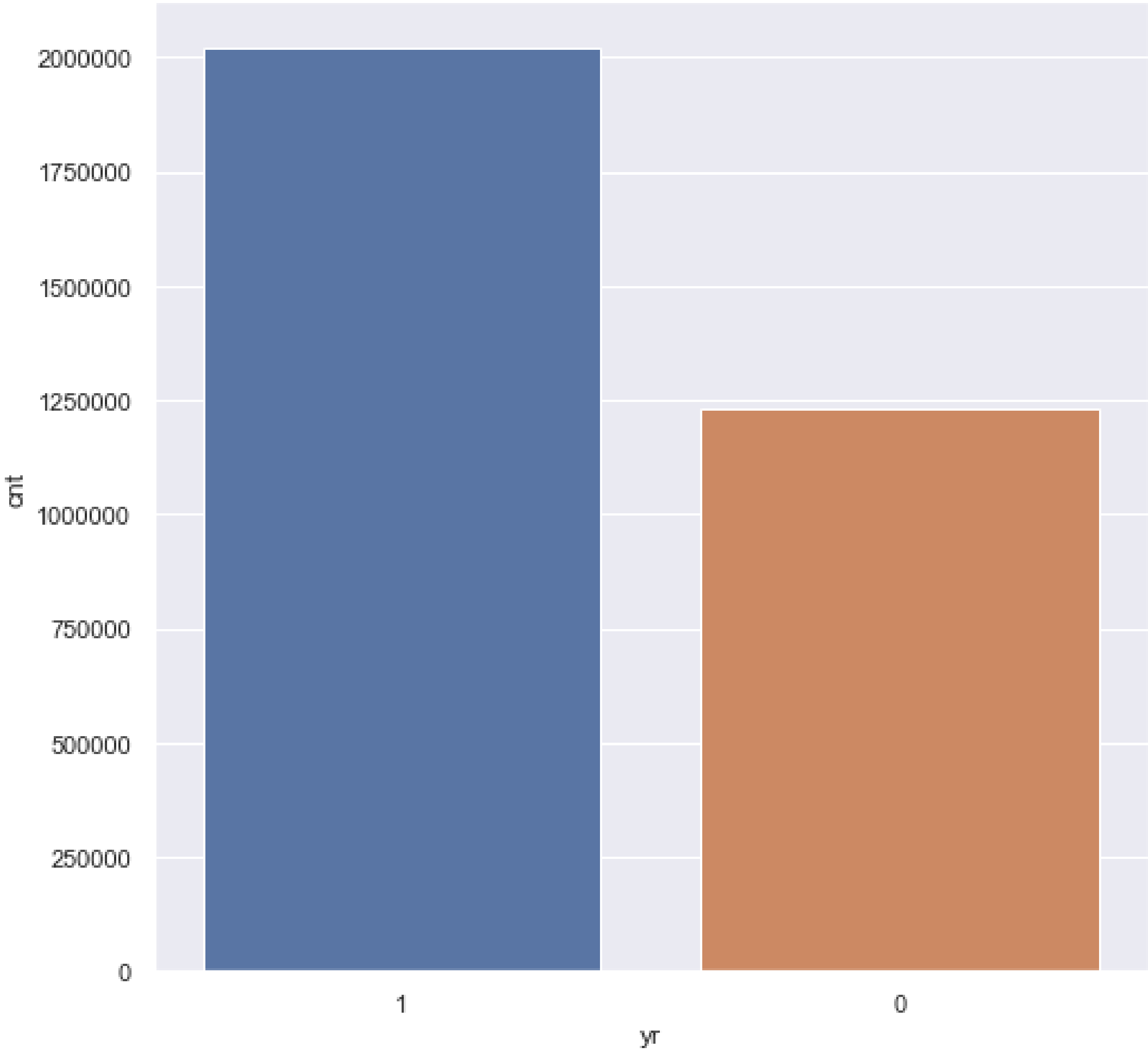
Workingday vs cnt



It can be seen that working day has significant effect on bike count .

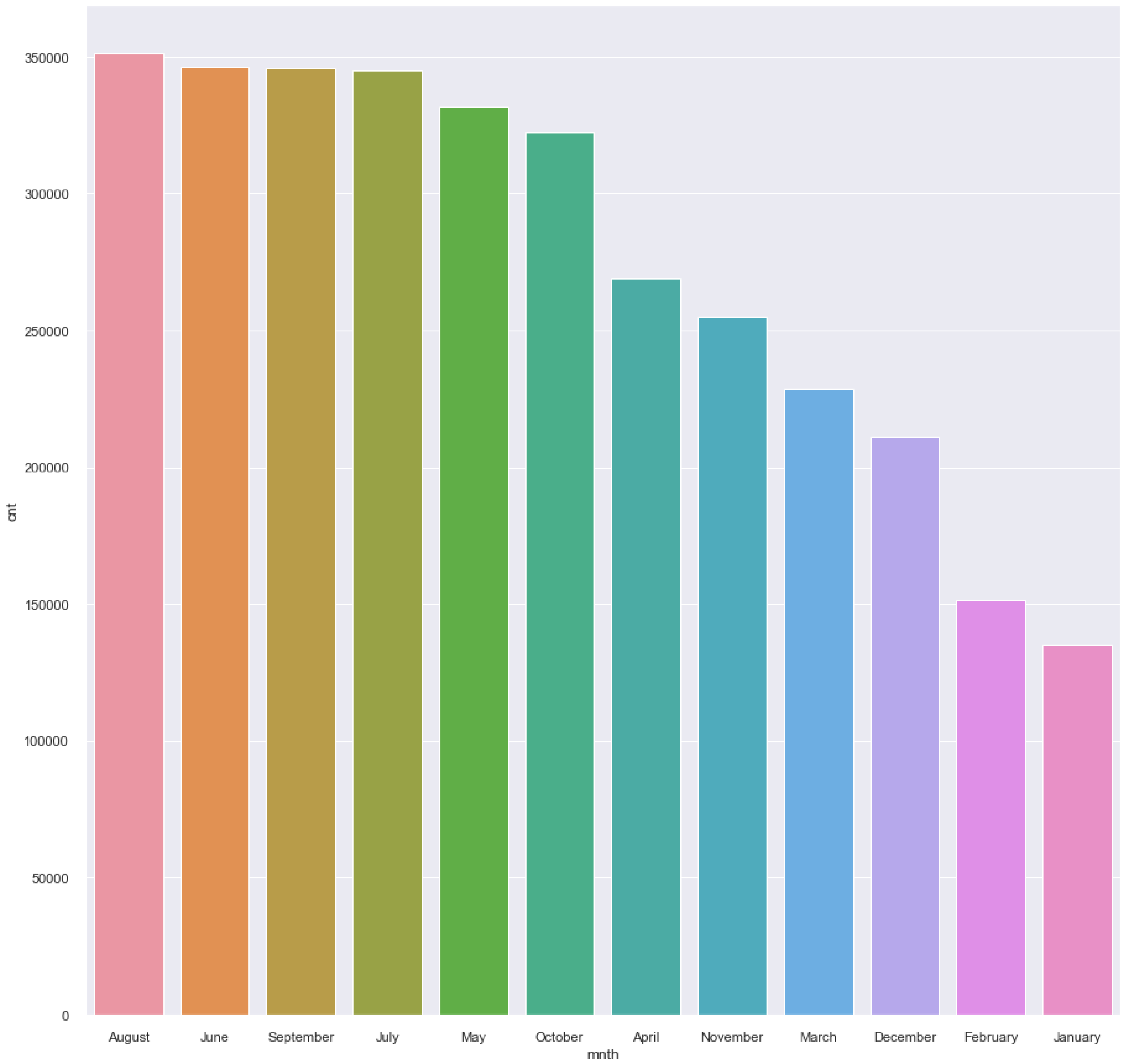
Yr vs bike count :

Rease



As seen from the graph that as year is progressing there is increase in demand of count.

Mnth vs cnt



it can be seen from above that august demand is max and in January its least so demand depends on mnth

Analysing Distribuition of numerical variables wrt to cnt by using regplot in seaborn:

Tmp vs cnt:

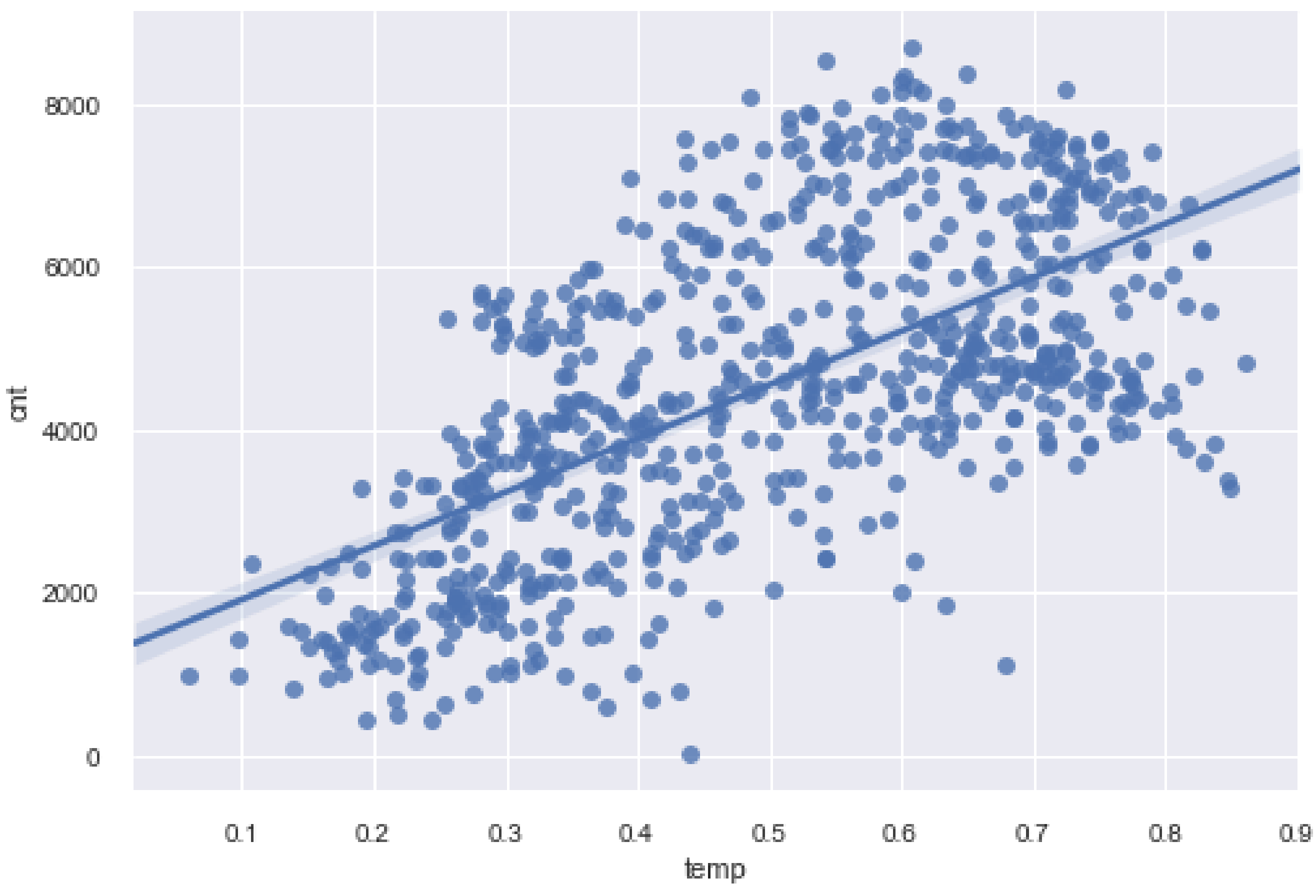
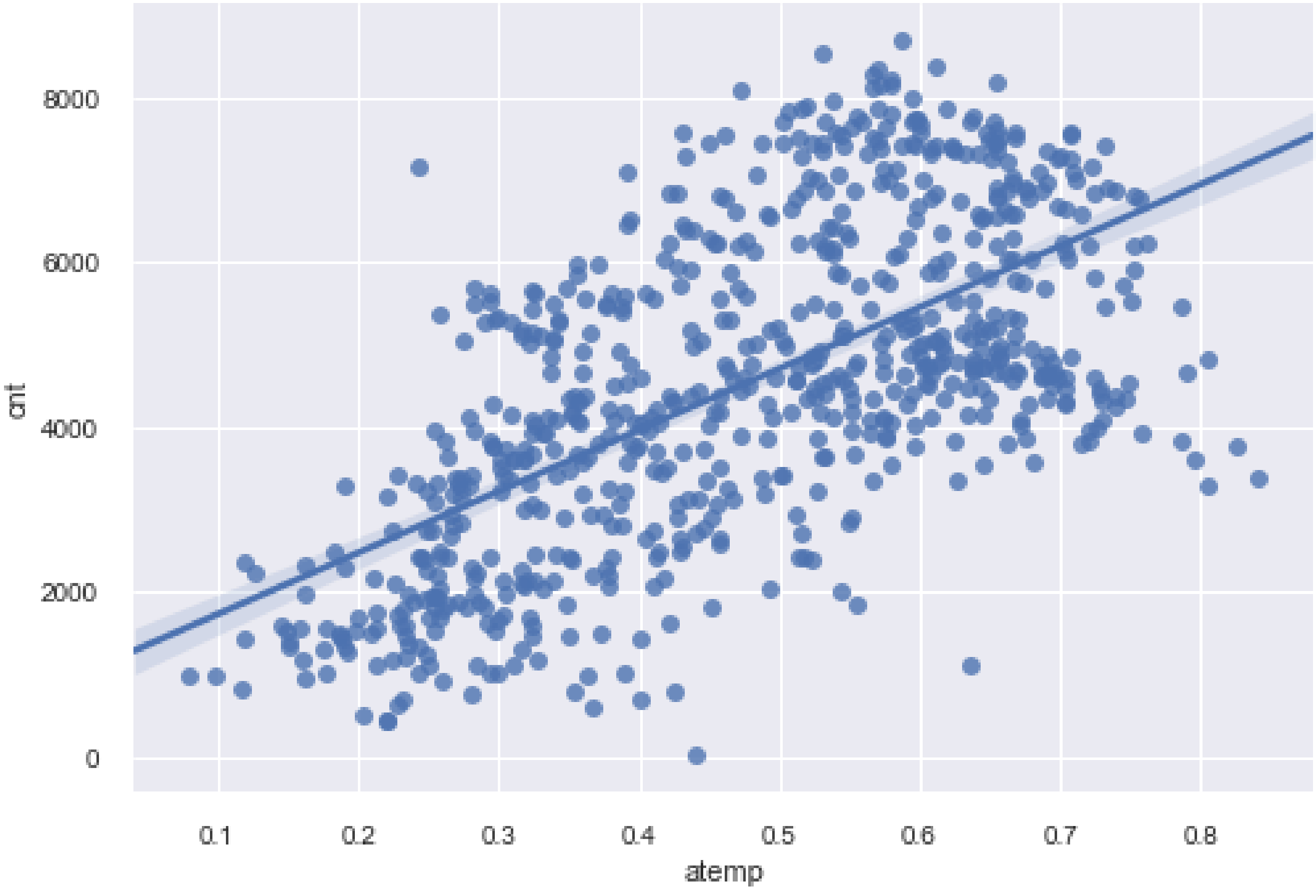
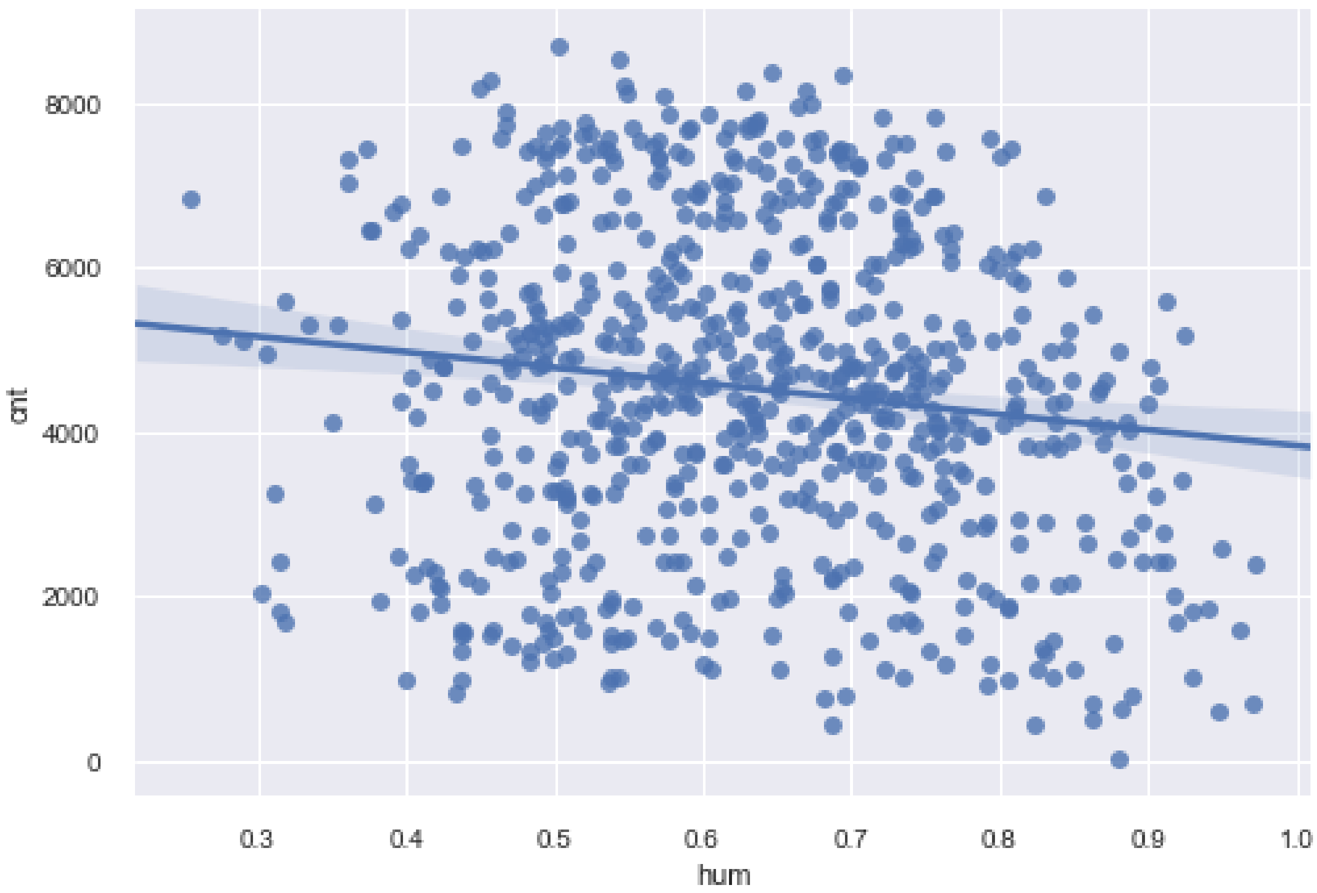
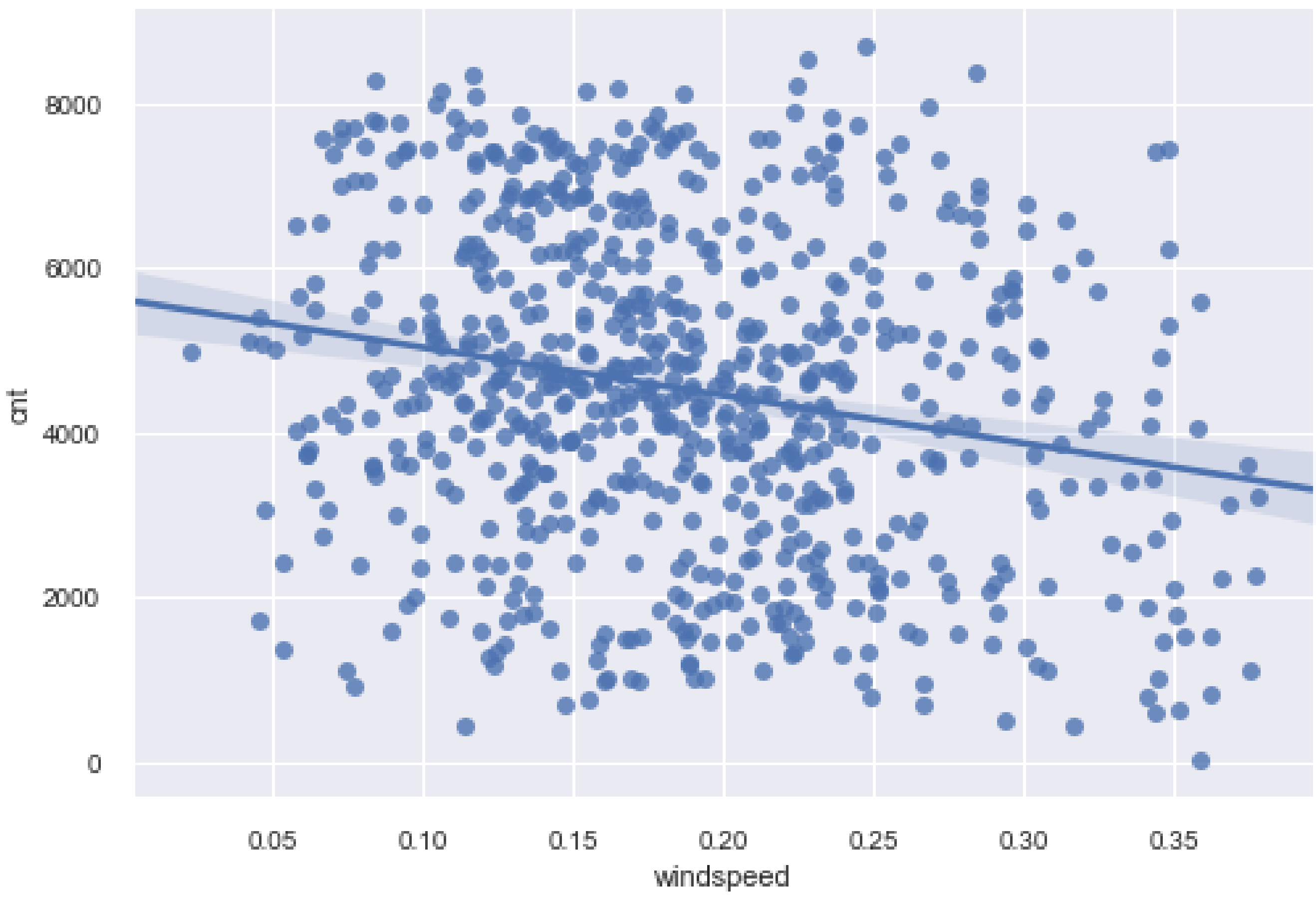

8000
6000
4000
2000
0
cnt
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
temp

Fig 2.3: temp vs cnt




8000
6000
4000
2000
0
cnt
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
hum



We can see both positive and negative relationship in numerical variable

Feature selection by boruta(r) and backward elimination and correlation chart

Feature Selection reduces the complexity of a model and makes it easier to interpret. And reduces overfitting
Correlation heat map plot is used to find out if there is any multicollinearity between variables. The highly collinear variables are dropped and then the model is executed.

Heatmap correlation:

Fig 2.7: Correlation plot of all the variables

It can be seen both temp and temp are highely correlated and one need to be dropped so we will drop atemp from data

Feature elimination using backward elimination:

Given an external estimator that assigns weights to features (e.g., the coefficients of a linear model), recursive feature elimination (`RFE`) is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through a `coef_` attribute or through a `feature_importances_` attribute. Then, the least important features are pruned from current set of features.That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

** source sklearn documentation

Number of columns are reduced after applying feature selection by backward elimination .

Feature remained are

```
['season', 'yr', 'weekday', 'workingday', 'weathersit', 'temp', 'windspeed']
```

three columns are dropped
feature selection by boruta using random forestclassifier

```
['season', 'yr', 'mnth', 'weathersit', 'temp', 'hum', 'windspeed']
```

Chapter 3: Modelling

3.1 Model Selection

The dependent variable in our model is a continuous variable i.e., Count of bike rentals. Hence the models that we choose are Linear Regression and XGBOOST. The error metric chosen for the problem statement is Mean Absolute Error (MAE).

3.2 Multiple Linear Regression

Multiple linear regression is the most common form of linear regression analysis. Multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical.

OLS Regression Results

==

Dep. Variable: cnt R-squared (uncentered): 0.973

Model: OLS Adj. R-squared (uncentered): 0.972

Method: Least Squares F-statistic: 1109.

Date: Sun, 22 Dec 2019 Prob (F-statistic): 0.00

Time: 20:37:33 Log-Likelihood: -3897.8

No. Observations: 480 AIC: 7826.

Df Residuals: 465 BIC: 7888.

Df Model: 15

Covariance Type: nonrobust

==
=====

	coef	std err	t	P>|t|	[0.025	0.975]
temp	5276.6355	309.319	17.059	0.000	4668.800	5884.471
windspeed	-1566.7061	446.161	-3.512	0.000	-2443.447	-689.965
season_2	1136.3215	137.798	8.246	0.000	865.537	1407.106
season_3	767.4346	180.092	4.261	0.000	413.540	1121.329
season_4	1486.5852	116.871	12.720	0.000	1256.925	1716.245
yr_1	2076.7407	75.341	27.565	0.000	1928.689	2224.792
weekday_1	-167.9462	220.815	-0.761	0.447	-601.866	265.973
weekday_2	-111.1043	257.052	-0.432	0.666	-616.231	394.022
weekday_3	60.1695	256.513	0.235	0.815	-443.899	564.238
weekday_4	55.1560	252.607	0.218	0.827	-441.236	551.548
weekday_5	83.3596	254.061	0.328	0.743	-415.891	582.610
weekday_6	601.7719	136.837	4.398	0.000	332.877	870.667
workingday_1	667.2283	218.868	3.049	0.002	237.136	1097.321
weathersit_2	-562.6982	80.798	-6.964	0.000	-721.472	-403.924
weathersit_3	-2364.5275	228.828	-10.333	0.000	-2814.193	-1914.862

==
===

Omnibus: 95.207 Durbin-Watson: 1.983

Prob(Omnibus): 0.000 Jarque-Bera (JB): 237.975

Skew: -0.992 Prob(JB): 2.11e-52

Kurtosis: 5.821 Cond. No. 19.2

It can be seen that since cond.no is small and accuracy is good so linear regression model is one of good option

Out[218]:

```
Mape=0.1829491904514197
Rsquared,adjusted_rsquare=0.7944815869715298 0.7805323734175612
test_RMSE: 859.048476 train_RMSE: 813.592434

there is not much of difference between train and test rmse so it can be infere
ed that model is not overfitting
```

3.3 XGBOOST

It is an implementation of gradient boosting machines created by Tianqi Chen, now with contributions from many developers. It belongs to a broader collection of tools under the umbrella of the Distributed Machine Learning Community or DMLC who are also the creators of the popular mxnet deep learning library.

Using Xgboost, we can predict the value of bike count.RMSE train and test for this model was `RMSE: 644.448971 RMSE: 458.138063`. The MAPE for xgboost was 58%. Hence the accuracy for this model for test was 87 and train was 89percent

Chapter 4: Conclusion

We can compare the models using any of the following criteria:

1. Predictive Performance
2. Interpretability
3. Computational Efficiency

4.

4.1 Computational Efficiency

As xgboost works on gradient boosting and it has power of doing parallel computing so it is way ahead than linear regression.

Performance

4.2RMSE and adjusted

RMSE is one of measures used to calculate the predictive performance of the model. We will apply this measure to our models that we have generated in the previous section.

Linear Regression Model RMSE: RMSE: 859.048476
RMSE: 813.592434

XGBoost: RMSE: RMSE: 644.448971
RMSE: 458.138063

It can be seen that x boost is clear winner in case of RMSE

4.3 Rsquare:

R squared and adjusted r squared is one of
measures used to calculate the predictive
performance of the model. We will apply this
measure to our models that we have
generated in the previous section
Based on the above error metrics, Random Forest is the better model for our analysis.
Hence Random Forest is chosen as the model for prediction of bike rental count.

Linear model: 0.7944815869715298 0.7805323734175612

XGBoost: 0.8900378155185579 0.8798561317702762

Again XGBOOST IS CLEAR WInner

Chapter 5: Appendix

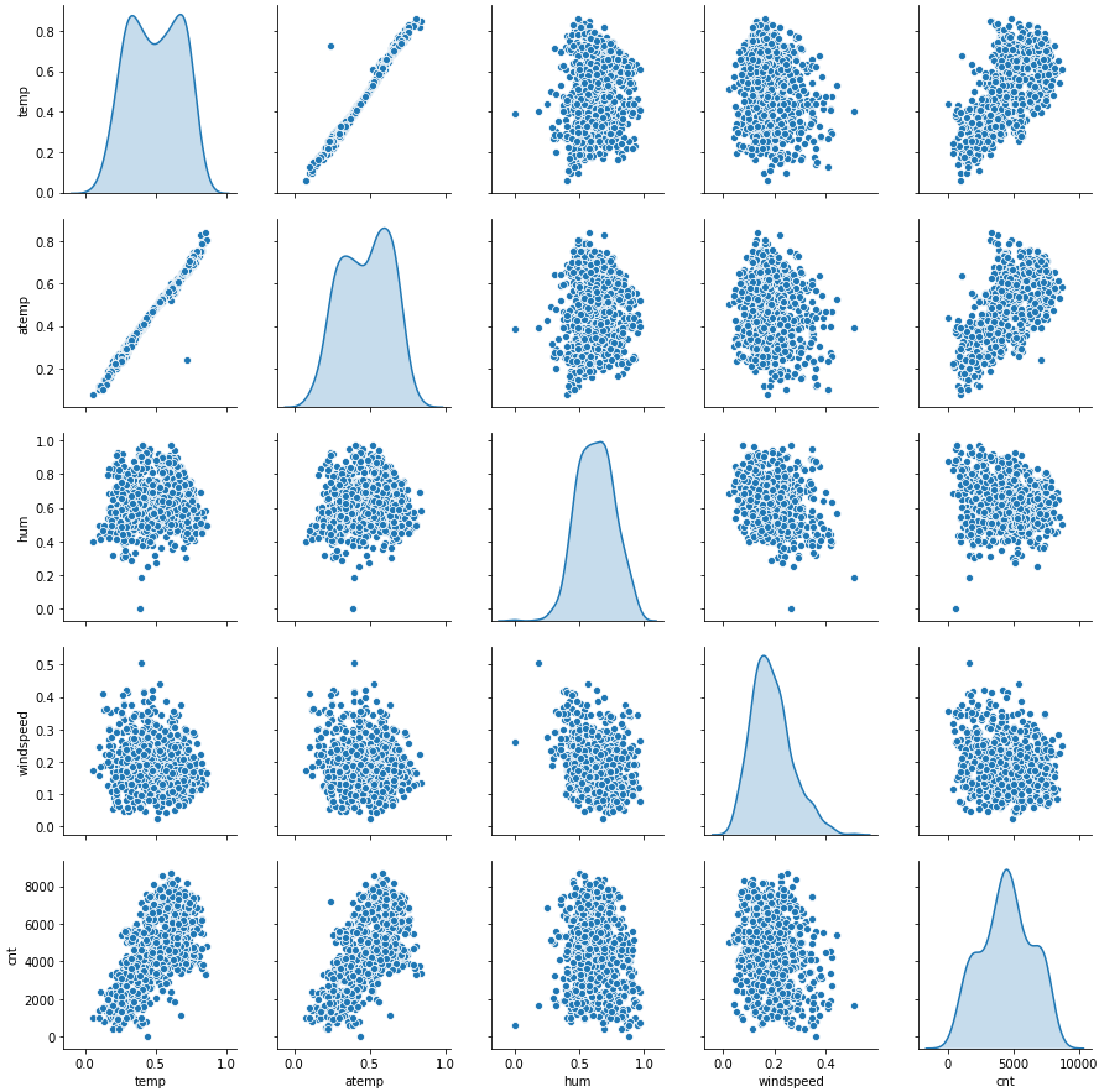
5.1 Figures

Fig 2.1: Distribution of continuous variables using Histograms

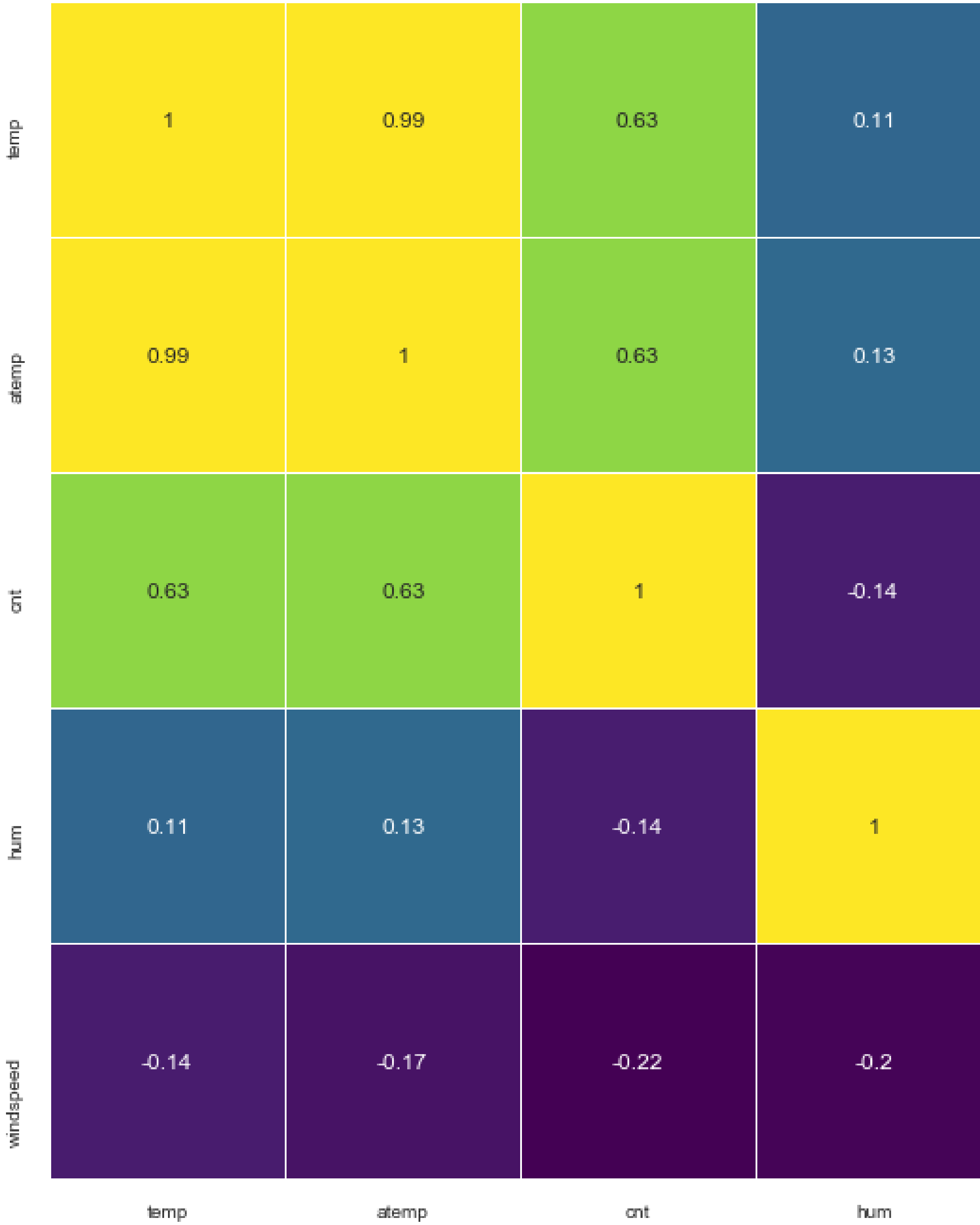
Fig 2.6: Distribution of numerical data using histograms after removal of outliers

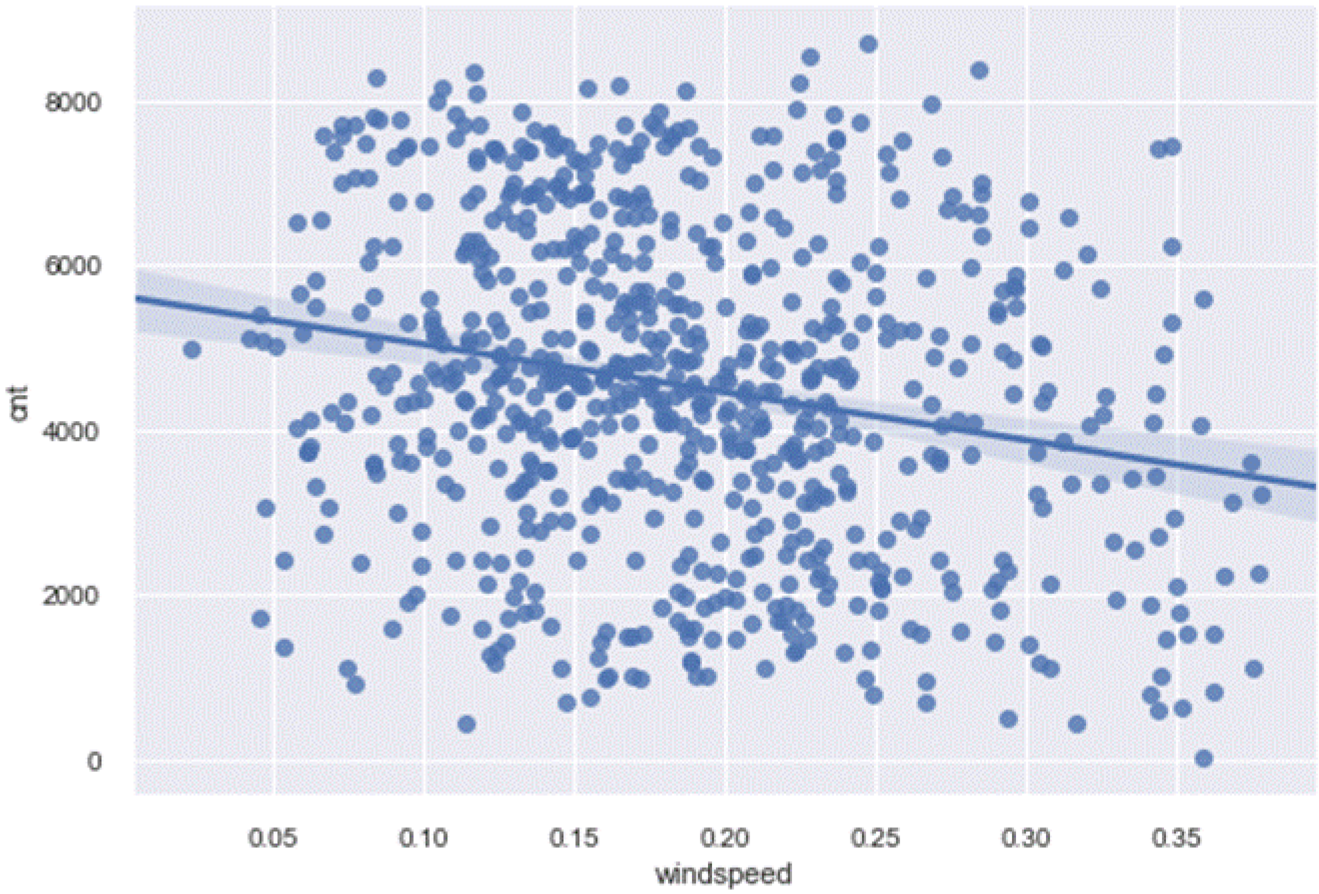
Fig 2.2: Distribution of categorical variables using bar plots

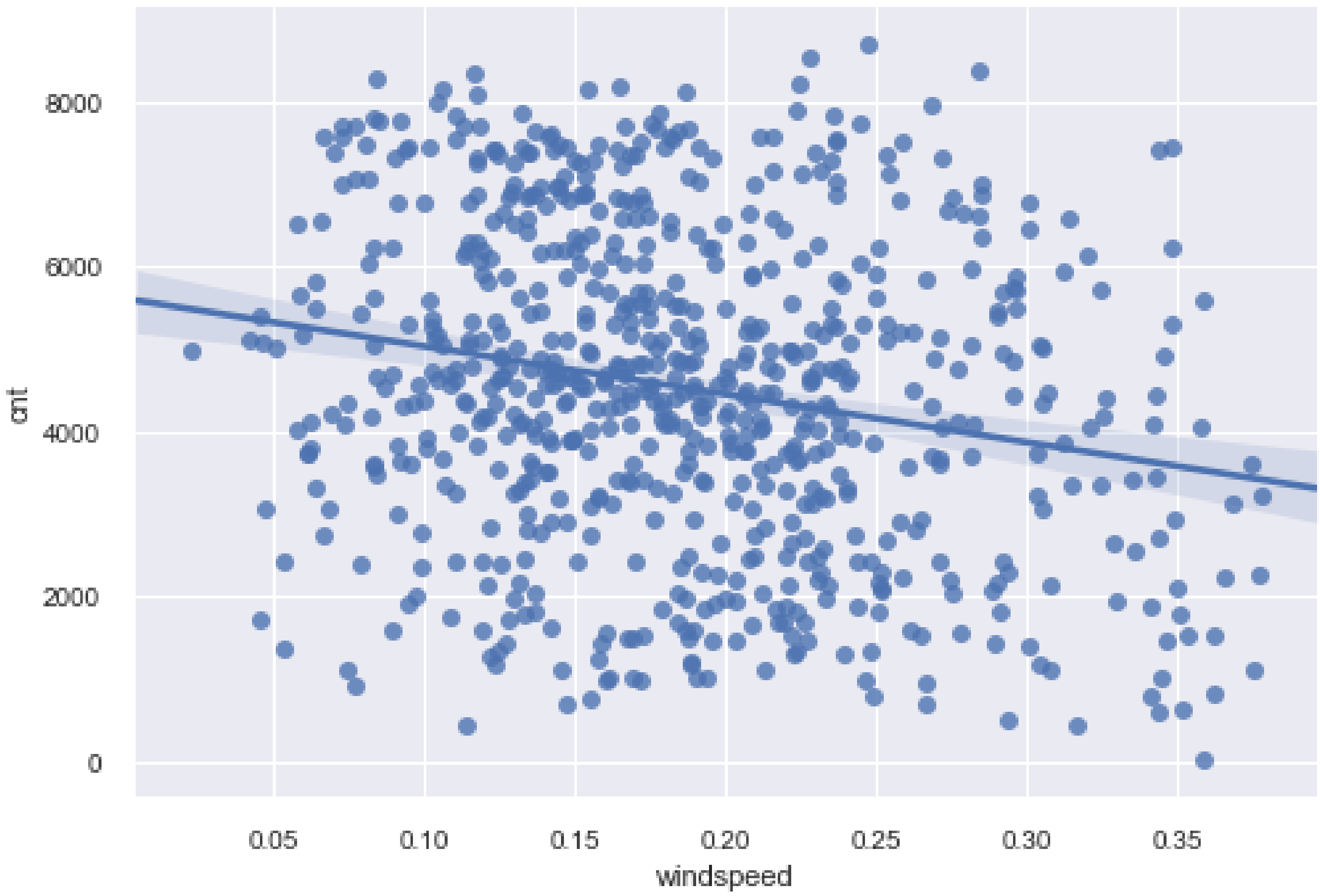
3: Scatter plot for continuous variables

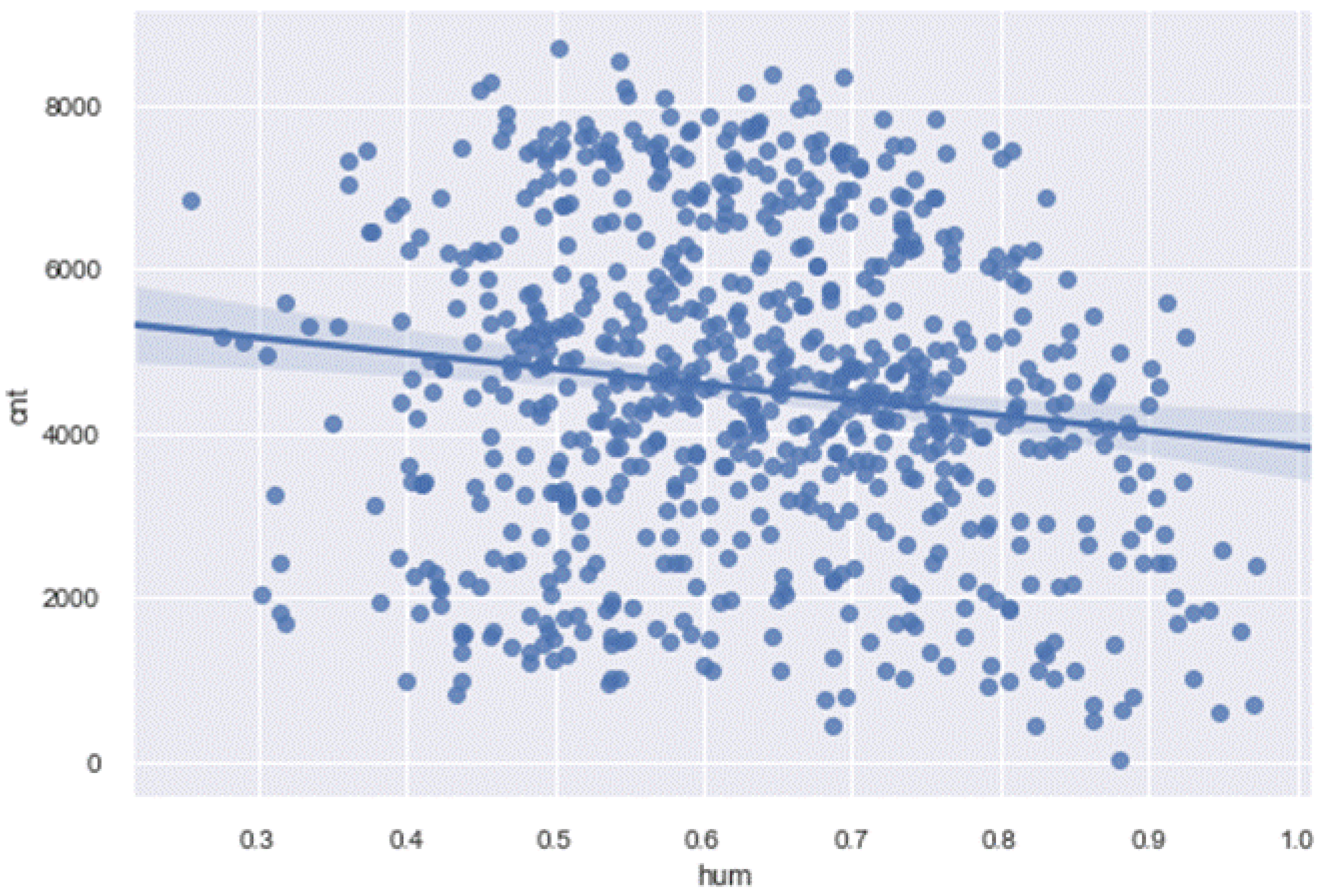


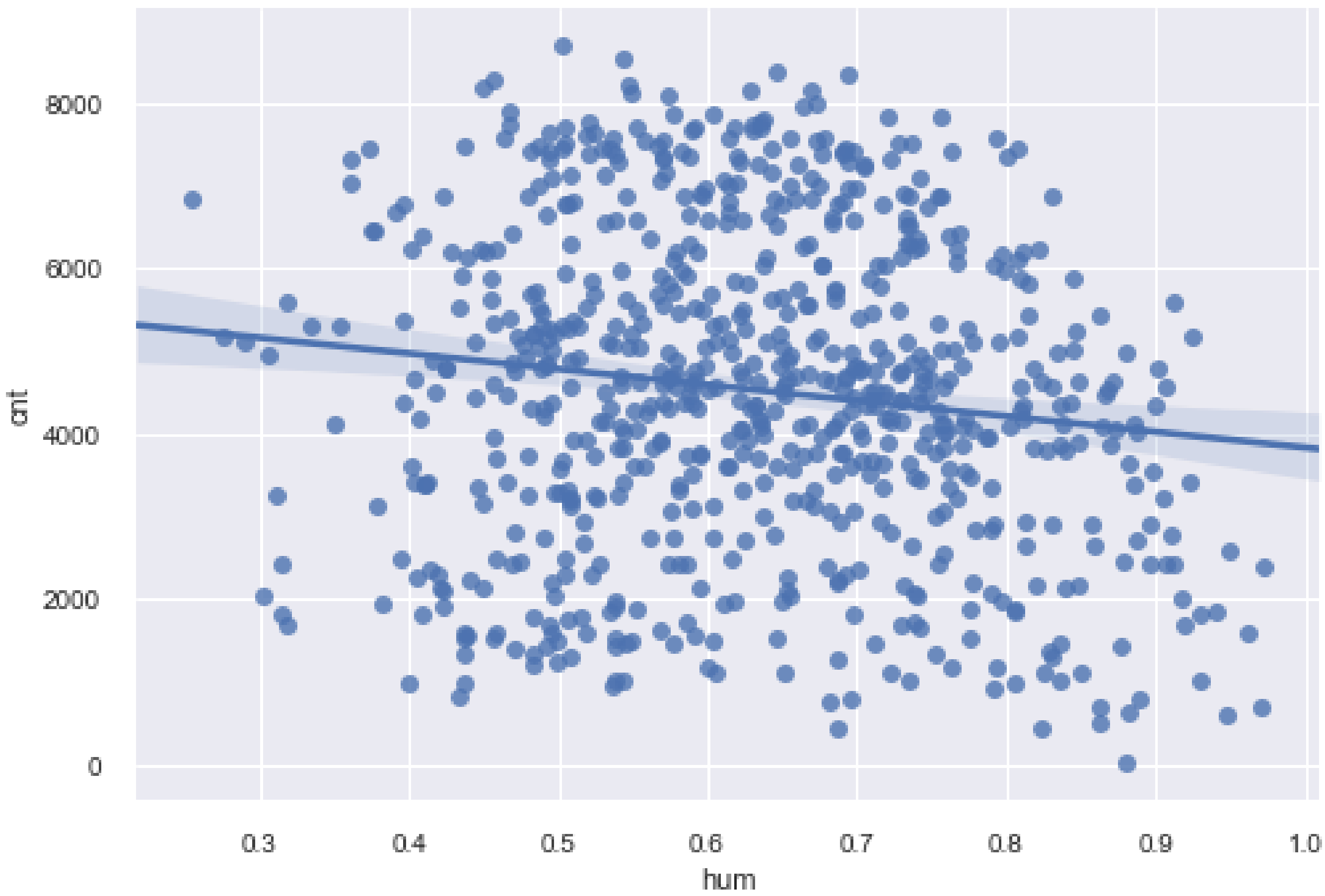
Scatter plot of all continuous variables

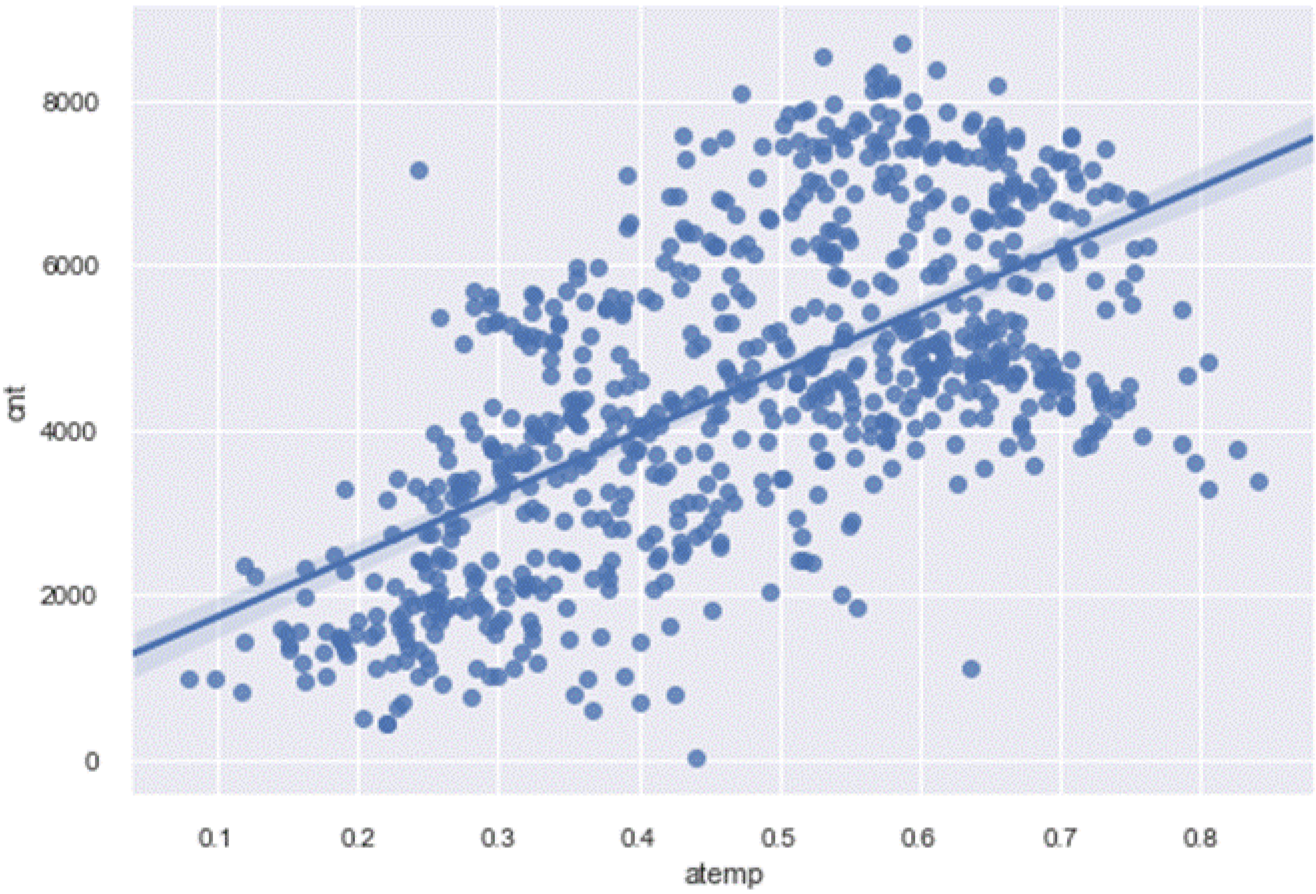

temp
atemp
cnt
hum
windspeed
1
0.99
0.63
0.11
0.99
1
0.63
0.13
0.63
0.63
1
-0.14
0.11
0.13
-0.14
1
-0.14
-0.17
-0.22
-0.2
temp
atemp
cnt
hum

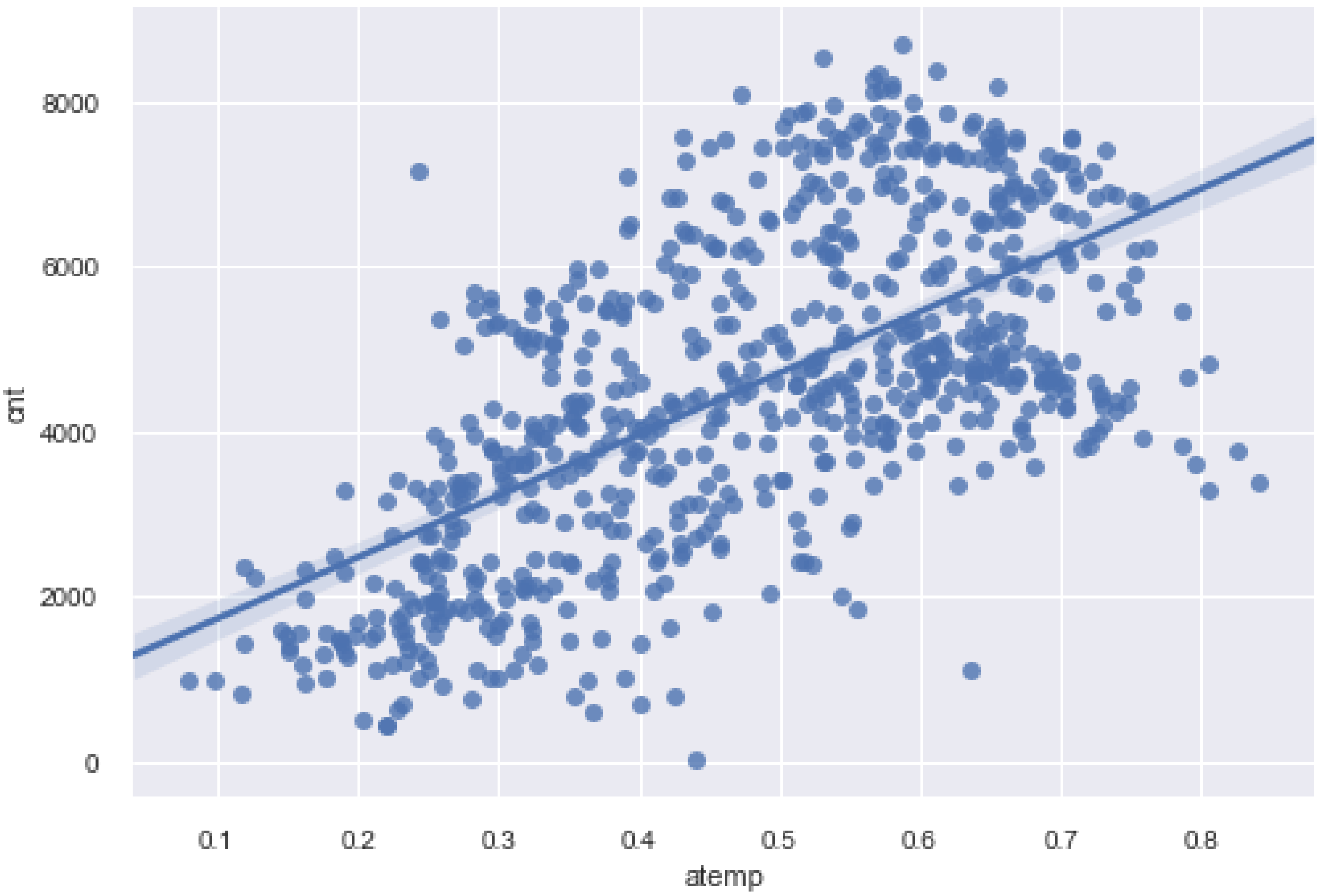

8000
6000
4000
2000
0
cnt
0.05
0.10
0.15
0.20
0.25
0.30
0.35
windspeed

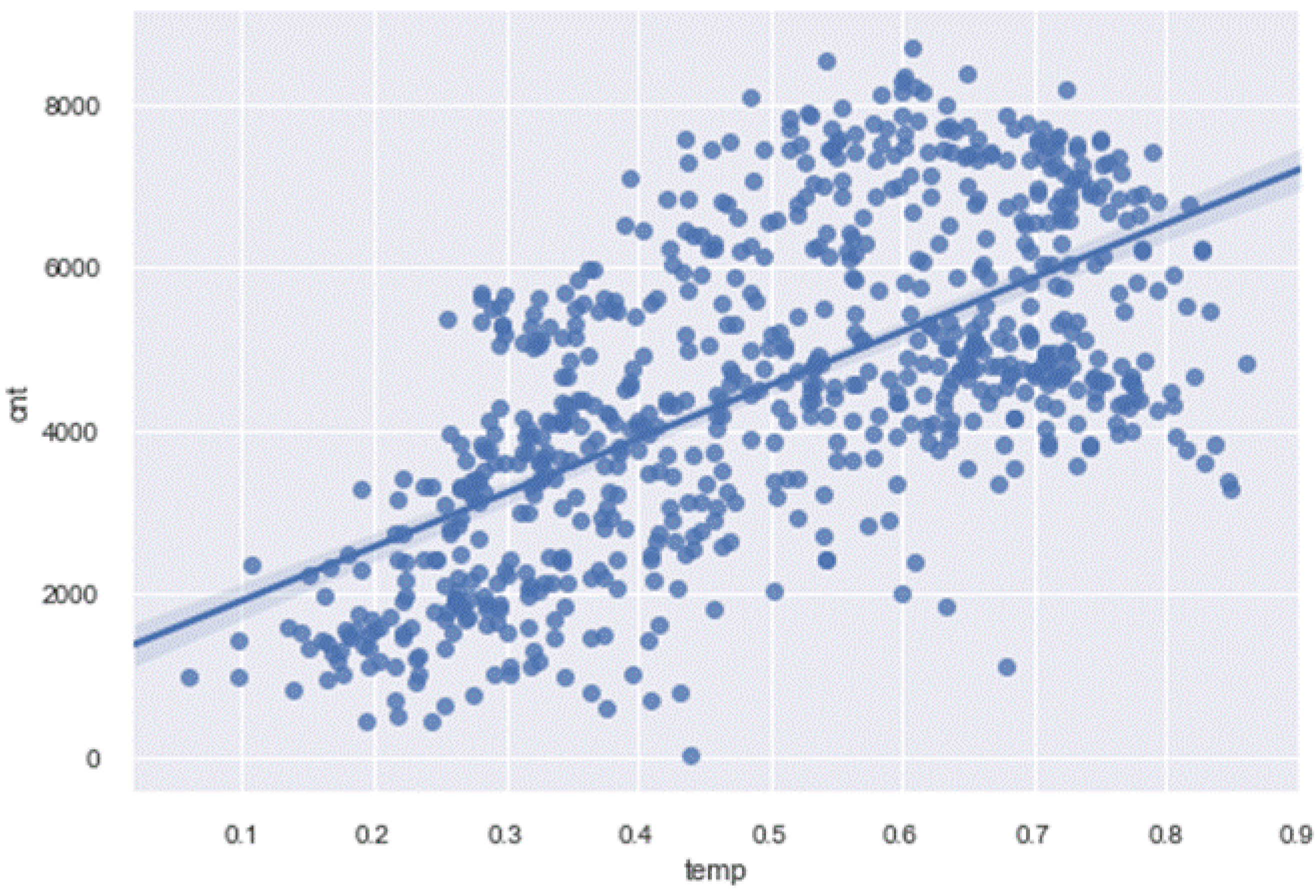

8000
6000
4000
2000
0
cnt
0.05
0.10
0.15
0.20
0.25
0.30
0.35
windspeed

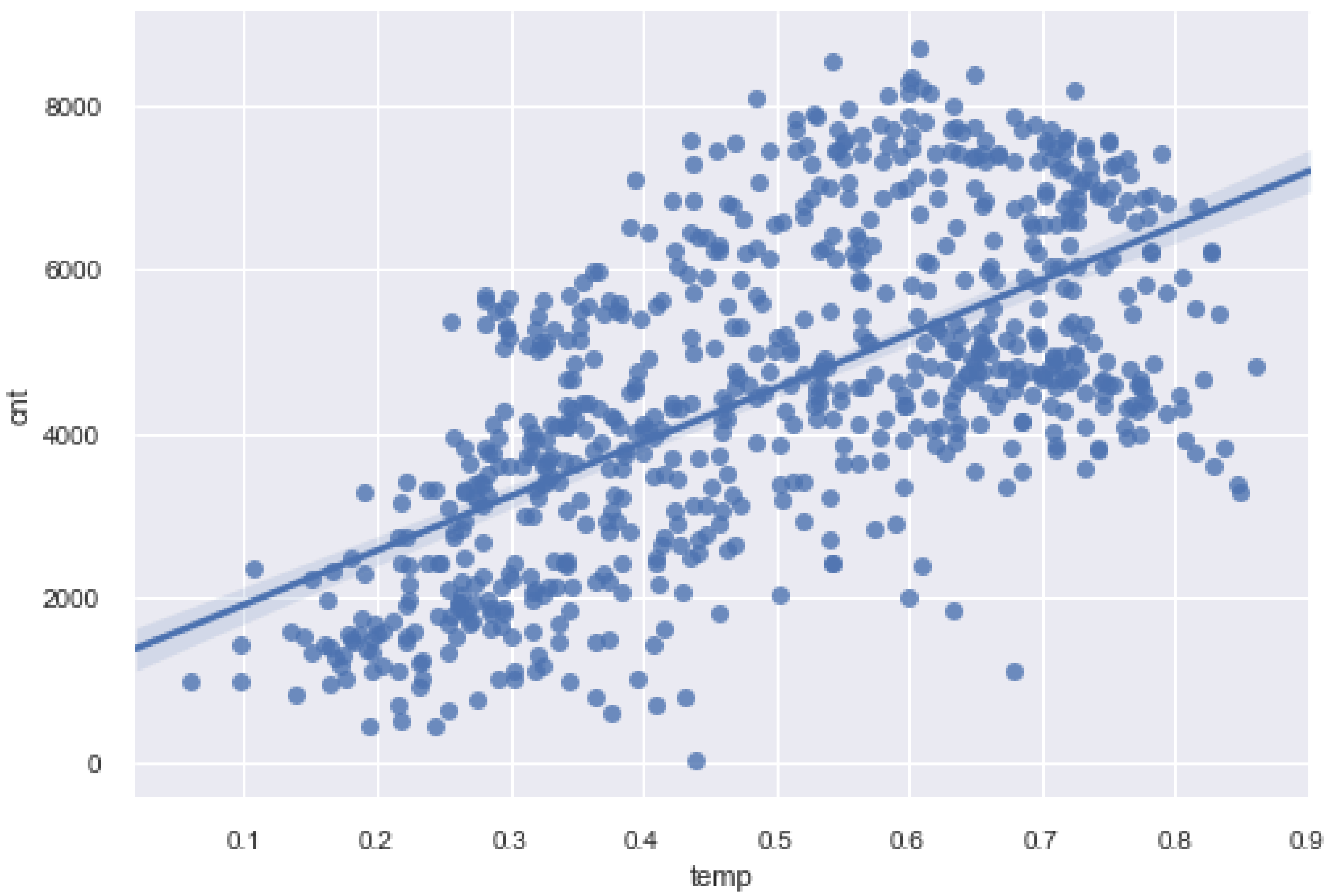

8000
6000
4000
2000
0
cnt
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
hum

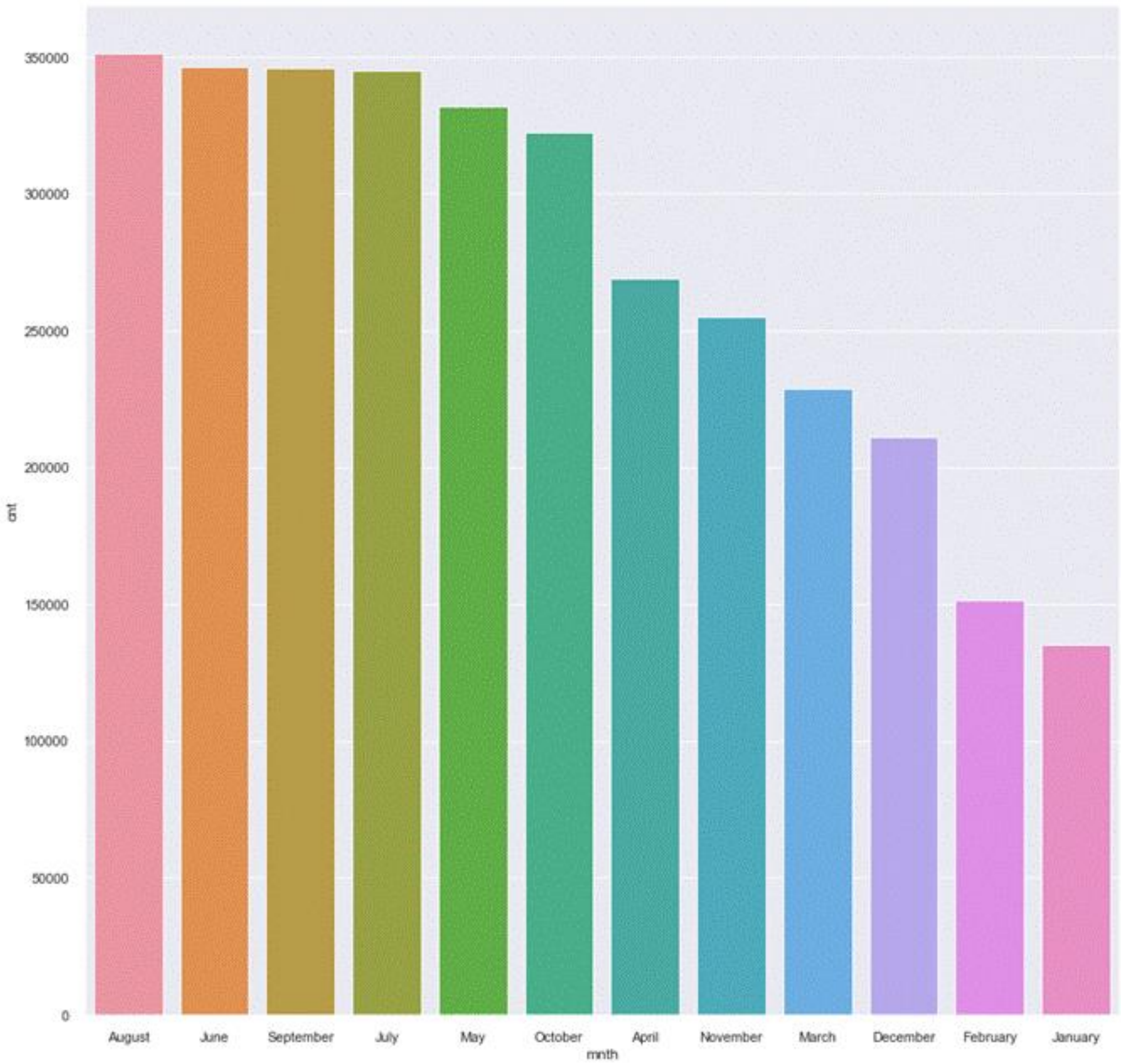

8000
6000
4000
2000
0
cnt
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
hum


8000
6000
4000
2000
0
cnt
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
atemp

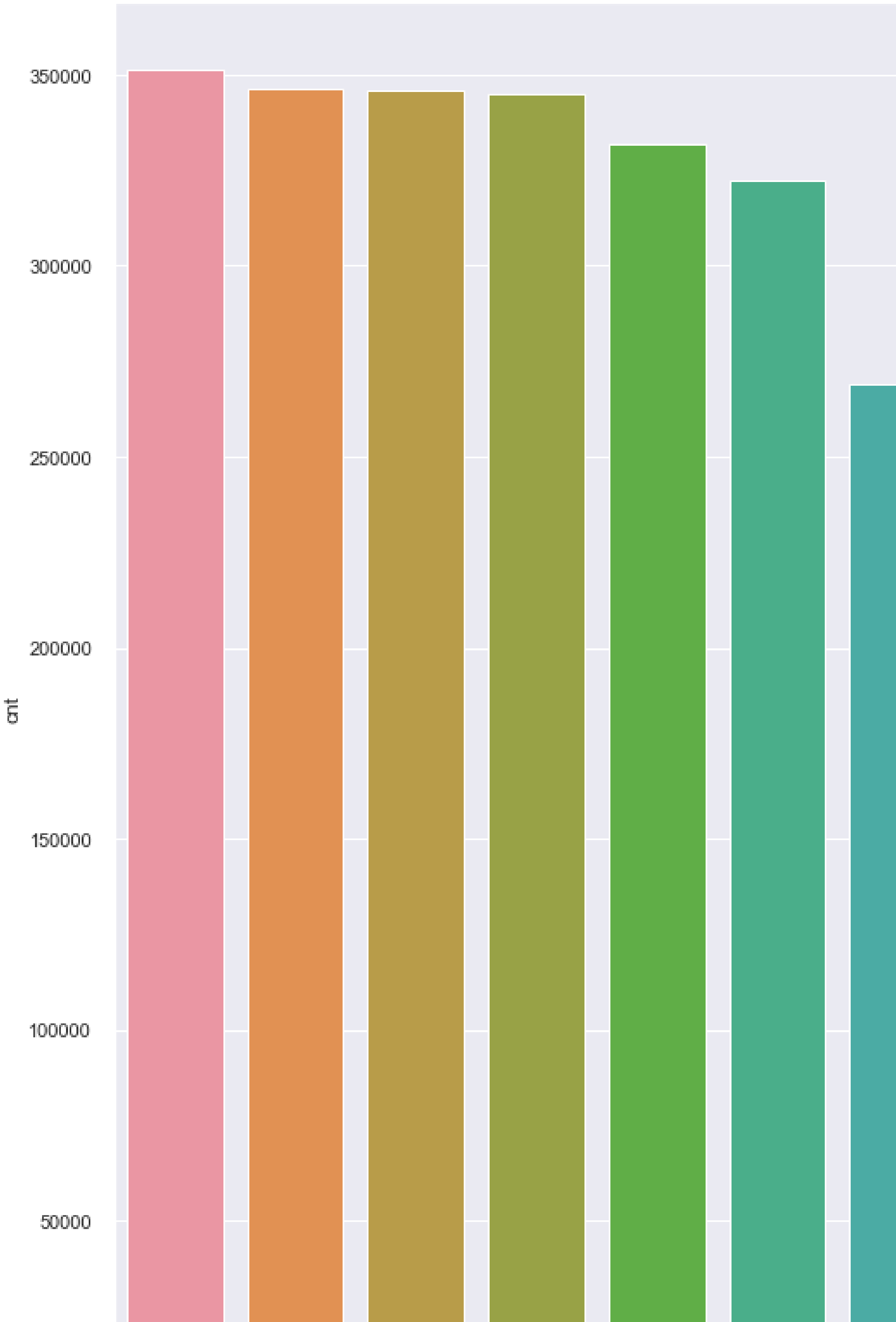

8000
6000
4000
2000
0
cnt
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
atemp


8000
6000
4000
2000
0
cnt
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
temp

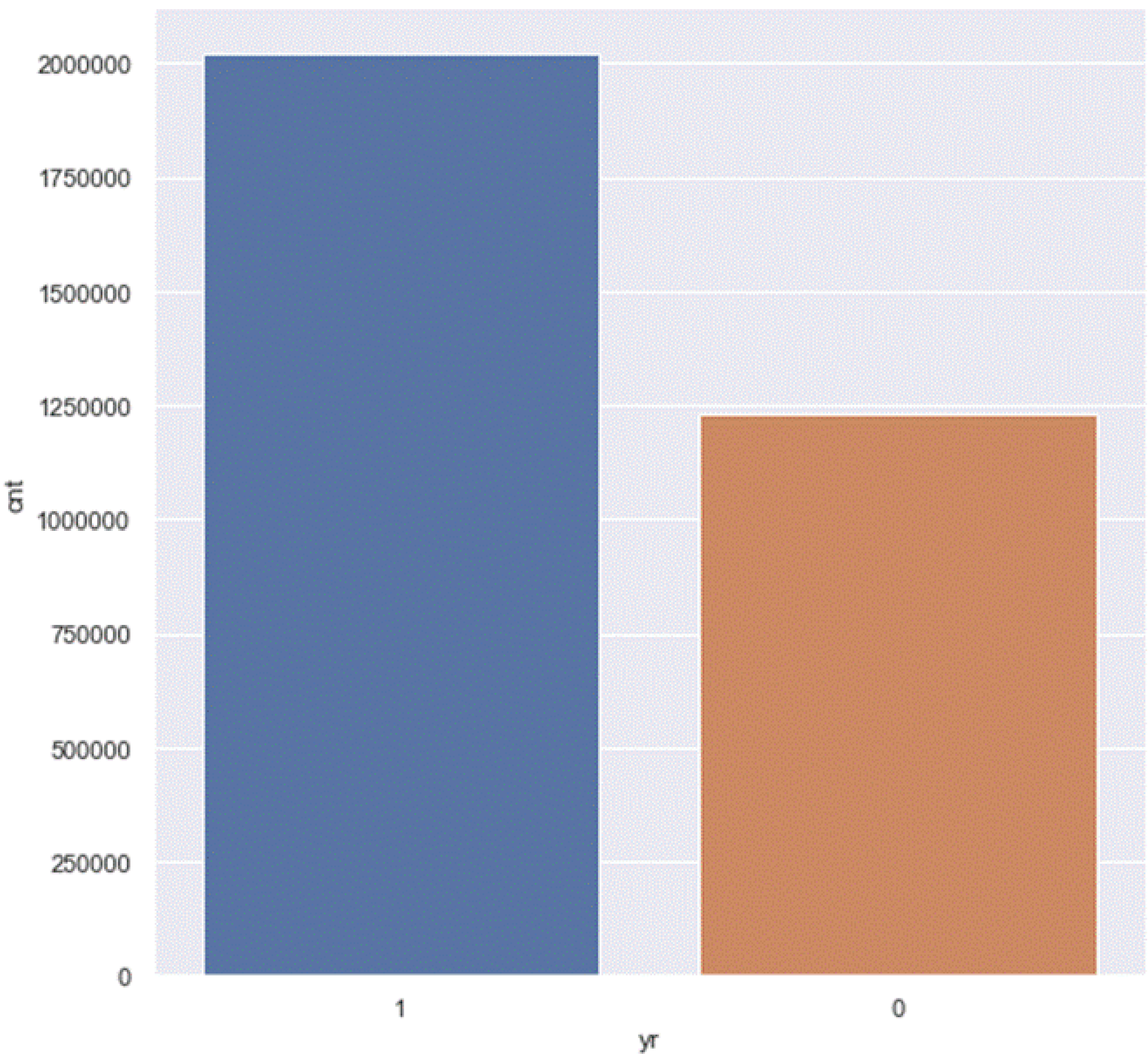

8000
6000
4000
2000
0
cnt
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
temp



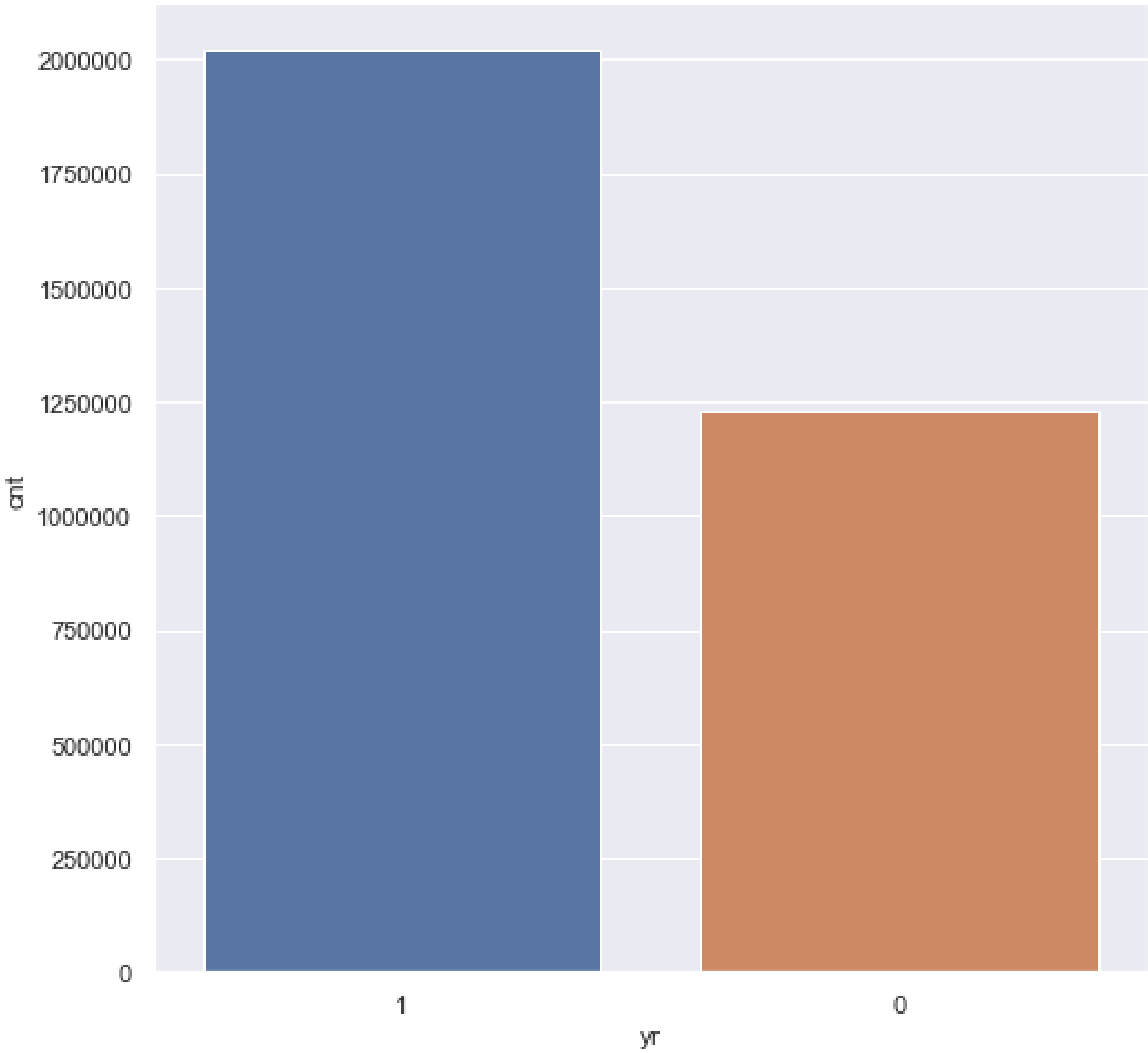
350000
300000
250000
200000
150000
100000
50000
0
cnt
August
June
September
July
May
October
April
November
March
December
February
January
mnth

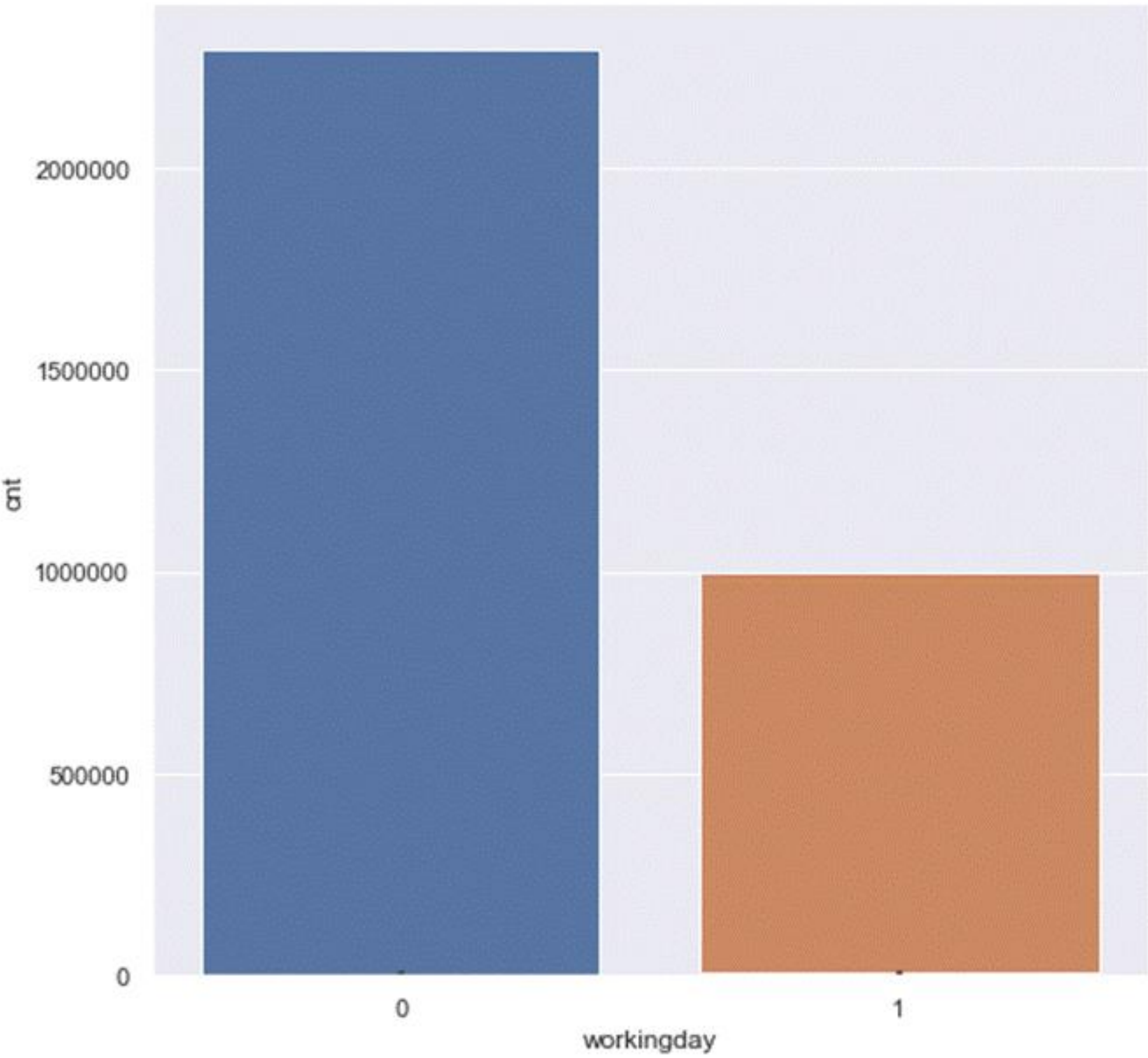


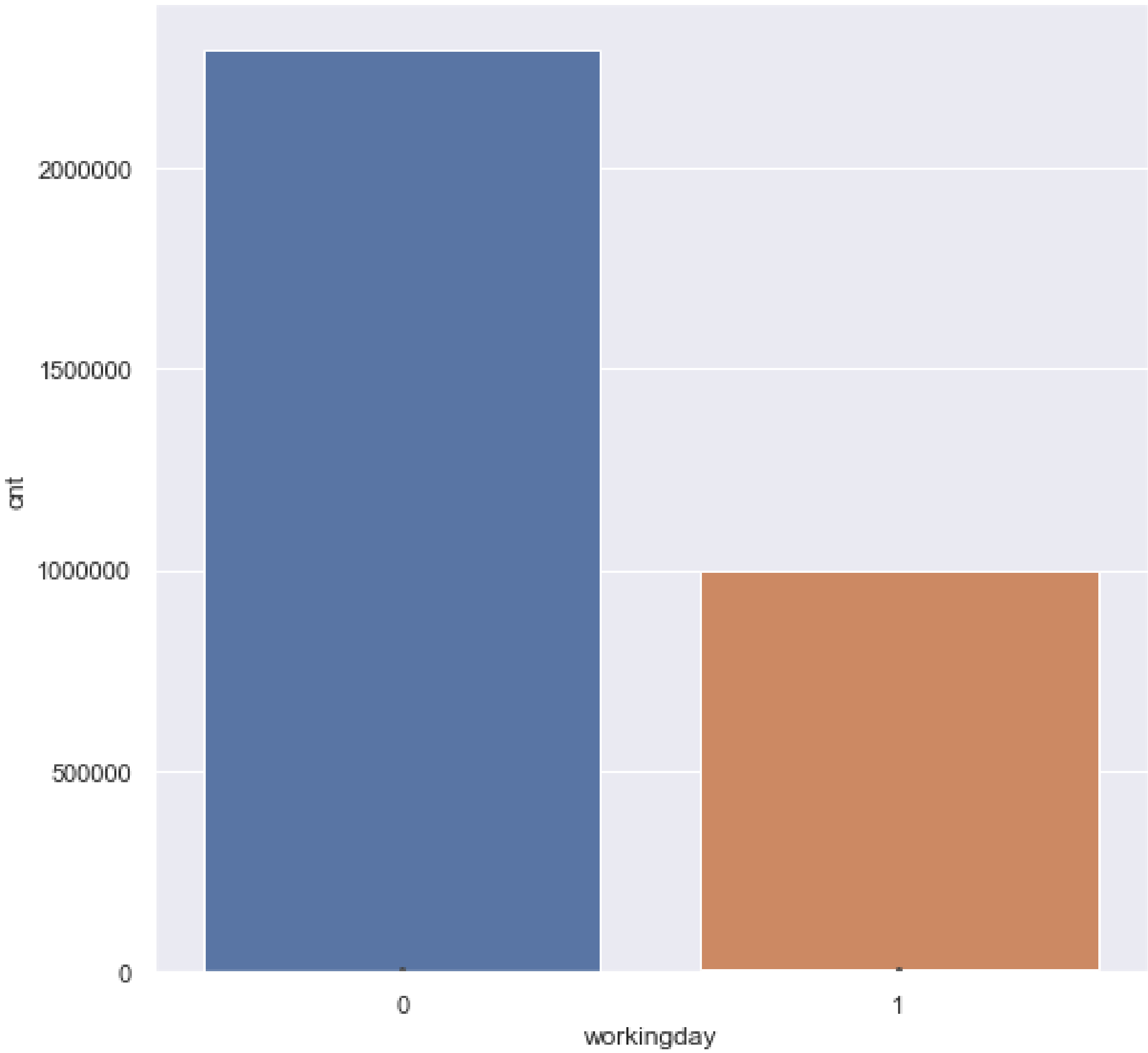
350000
300000
250000
200000
150000
100000
50000
cnt



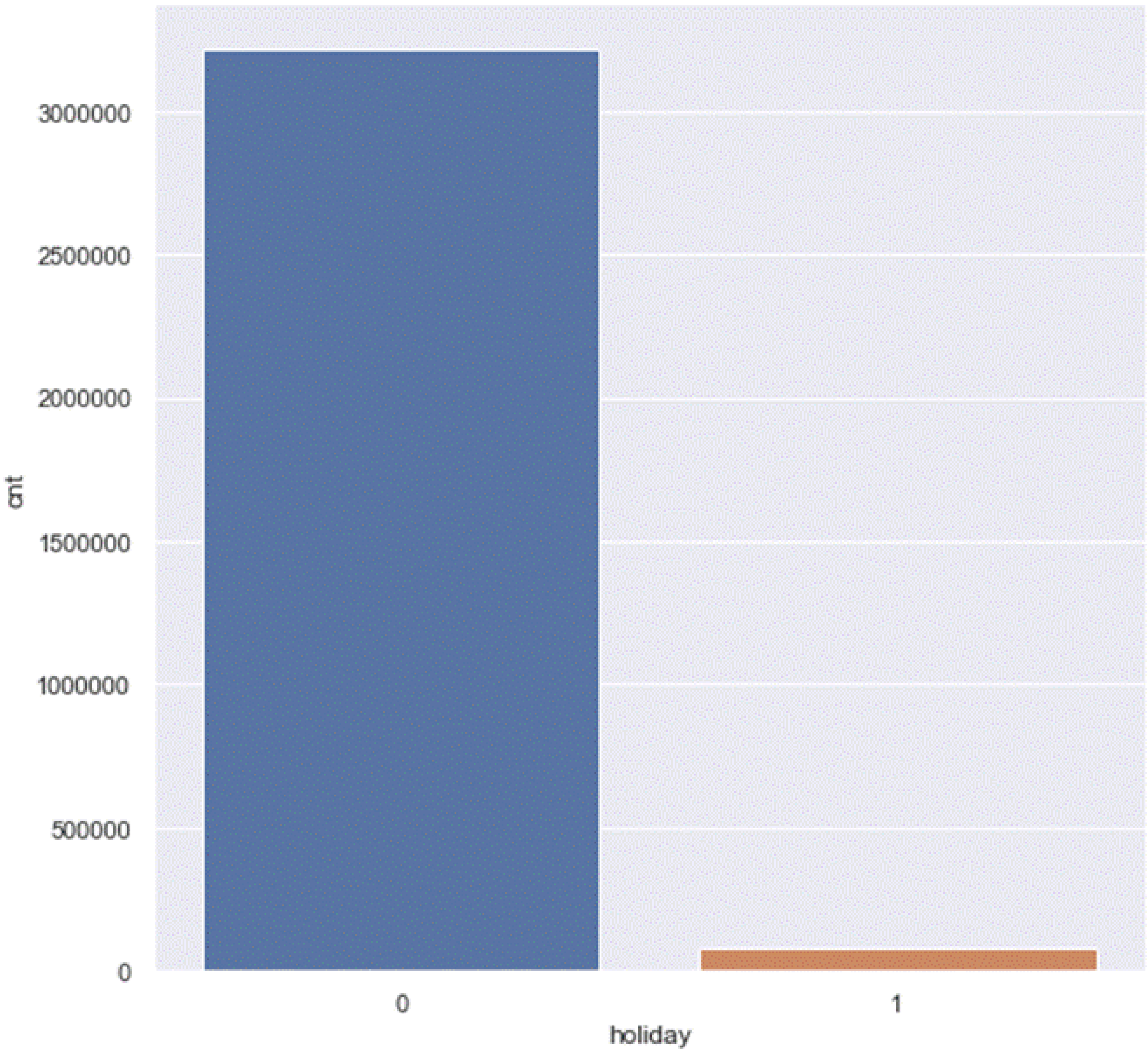
2000000
1750000
1500000
1250000
1000000
750000
500000
250000
0
cnt
1
0
yr


2000000
1750000
1500000
1250000
1000000
750000
500000
250000
0
cnt
1
0
yr

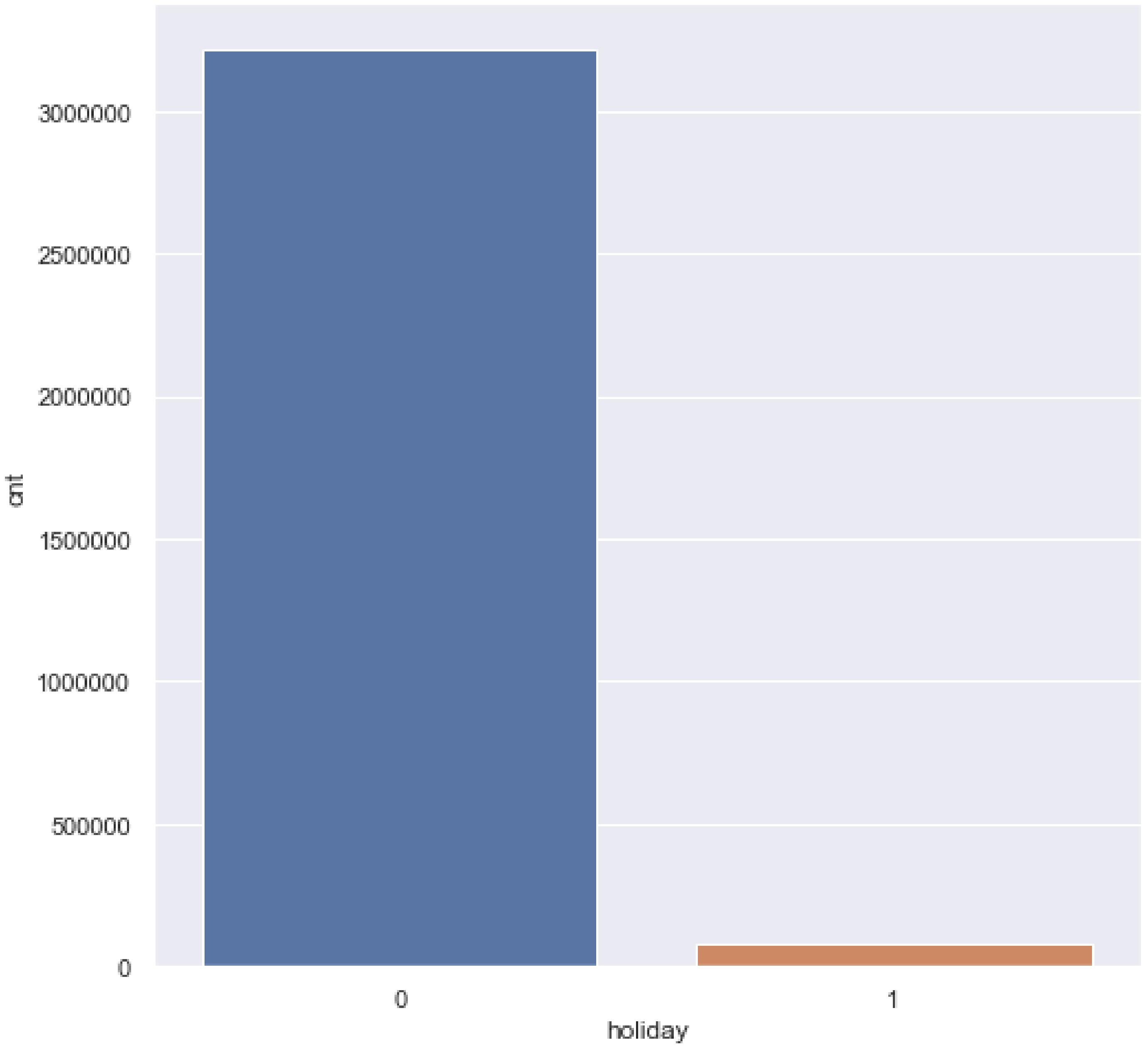

2000000
1500000
1000000
500000
0
cnt
0
1
workingday

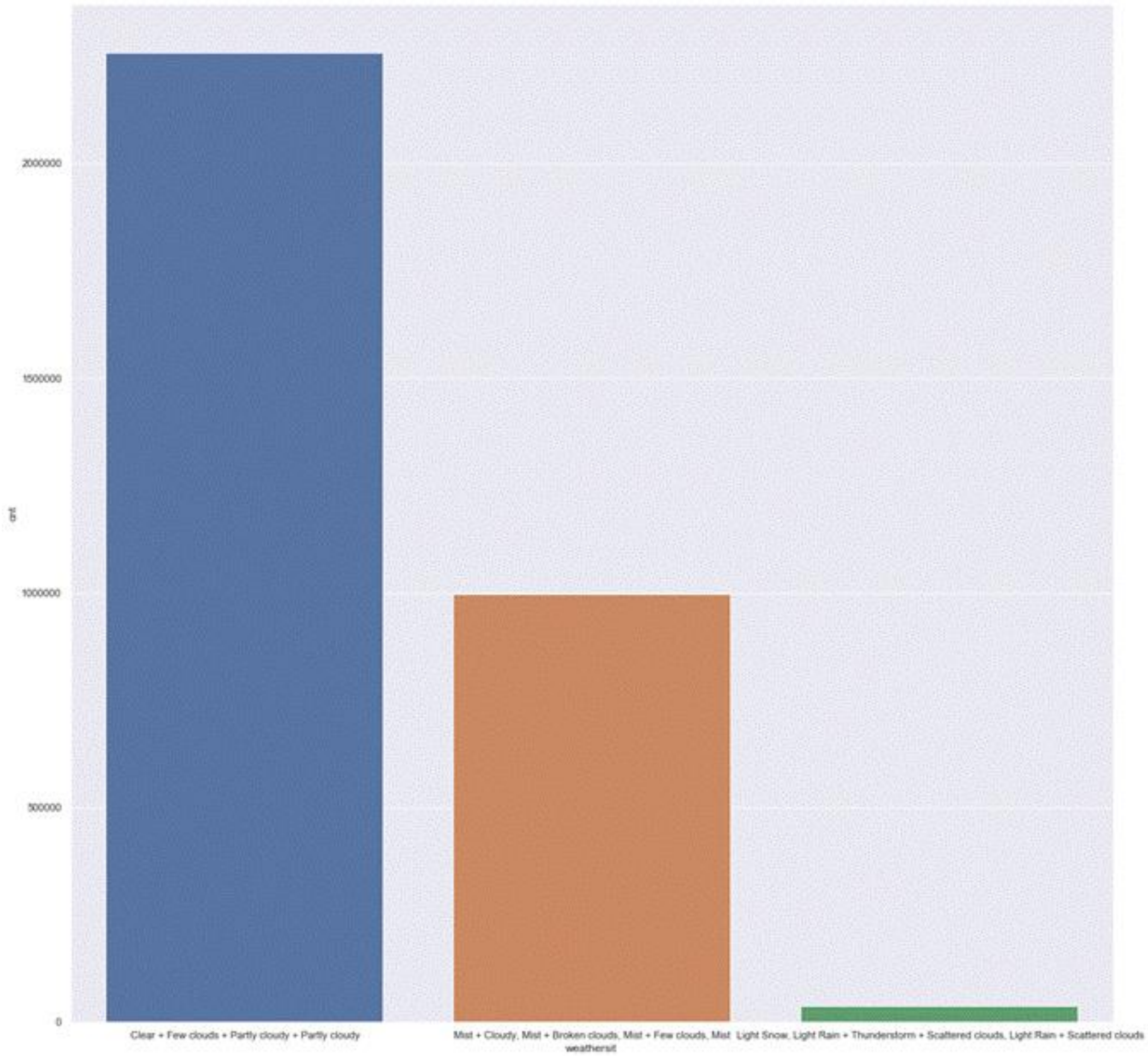


2000000
1500000
1000000
500000
0
cnt
0
1
workingday

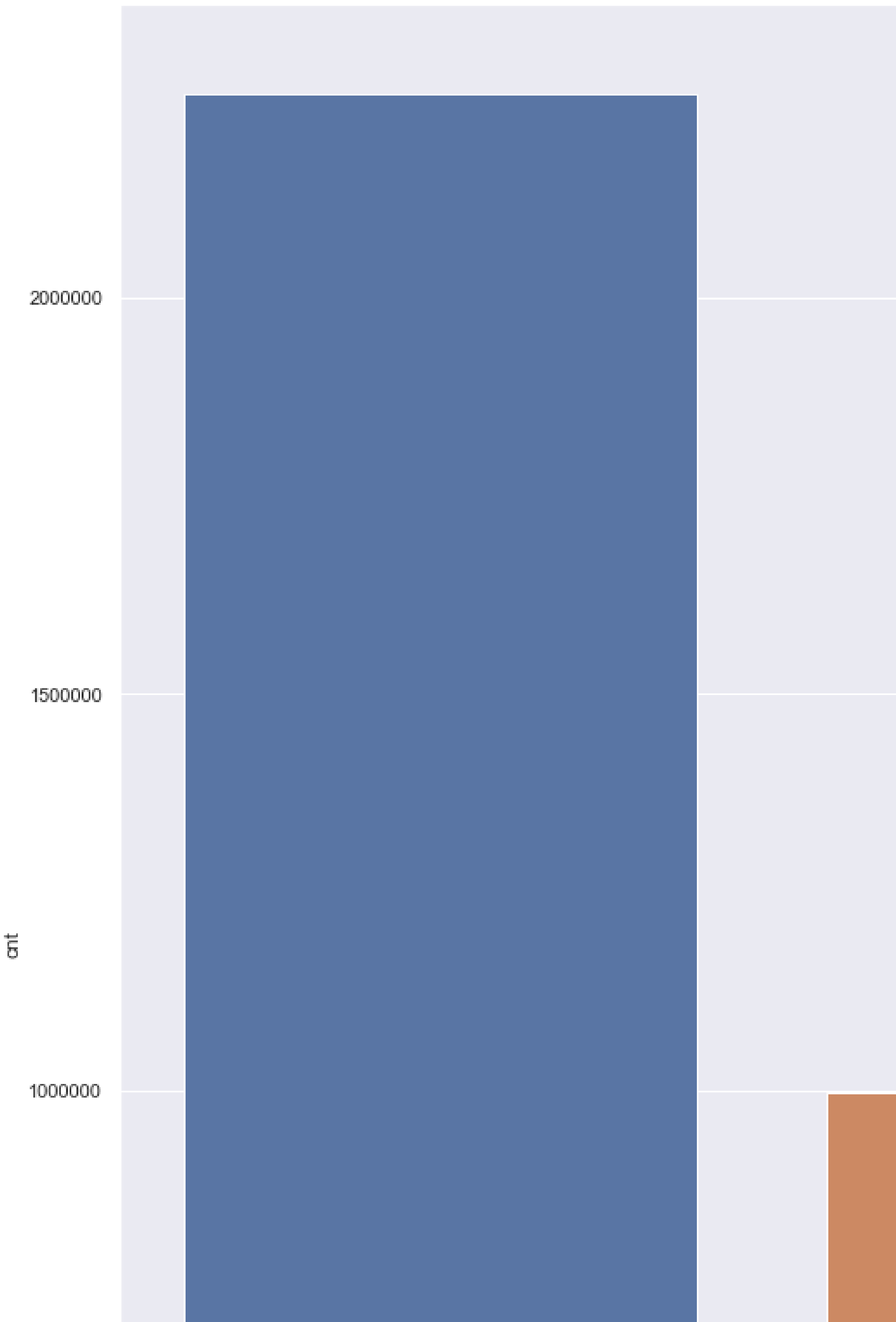


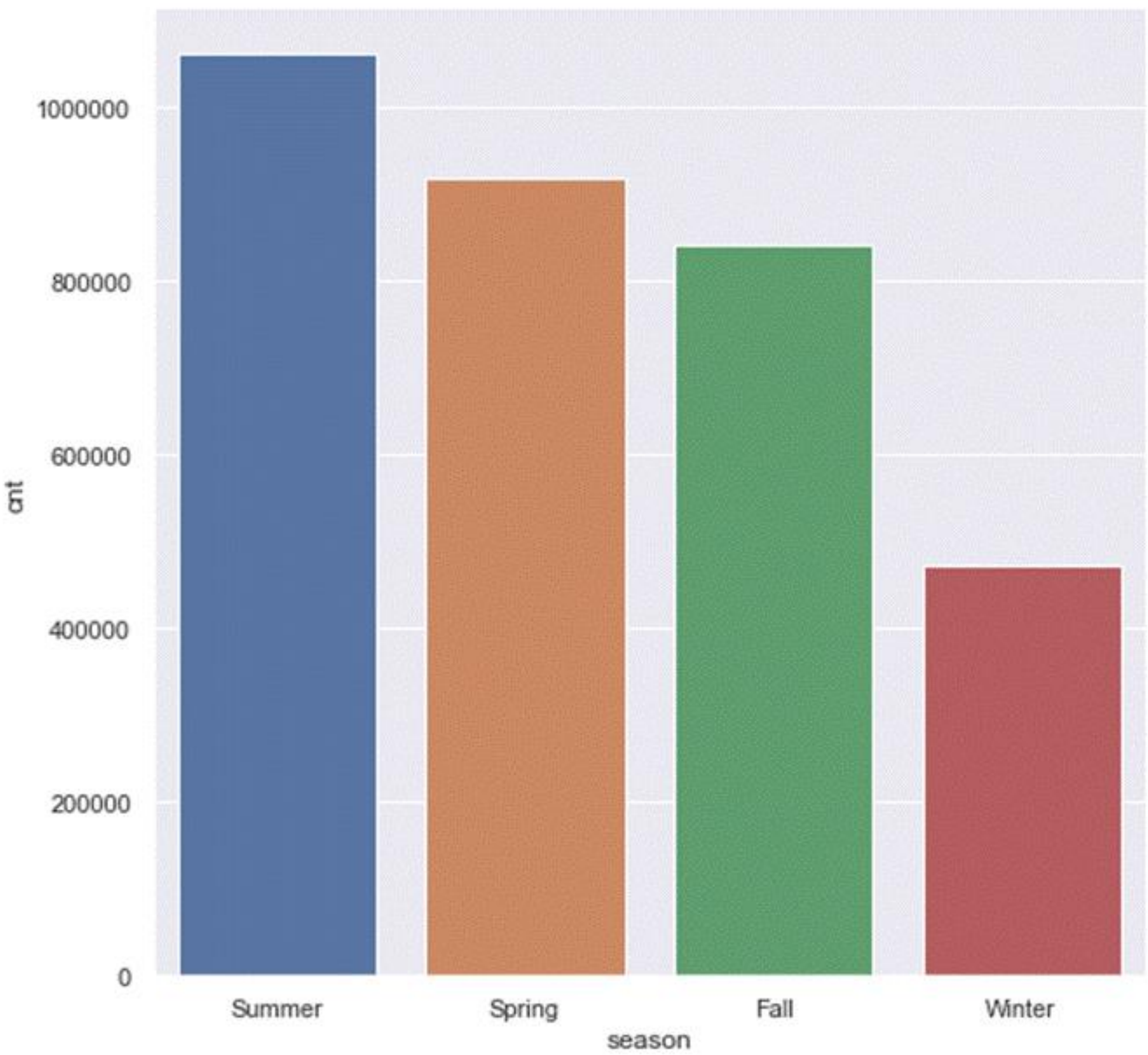
3000000
2500000
2000000
1500000
1000000
500000
0
cnt
0
1
holiday

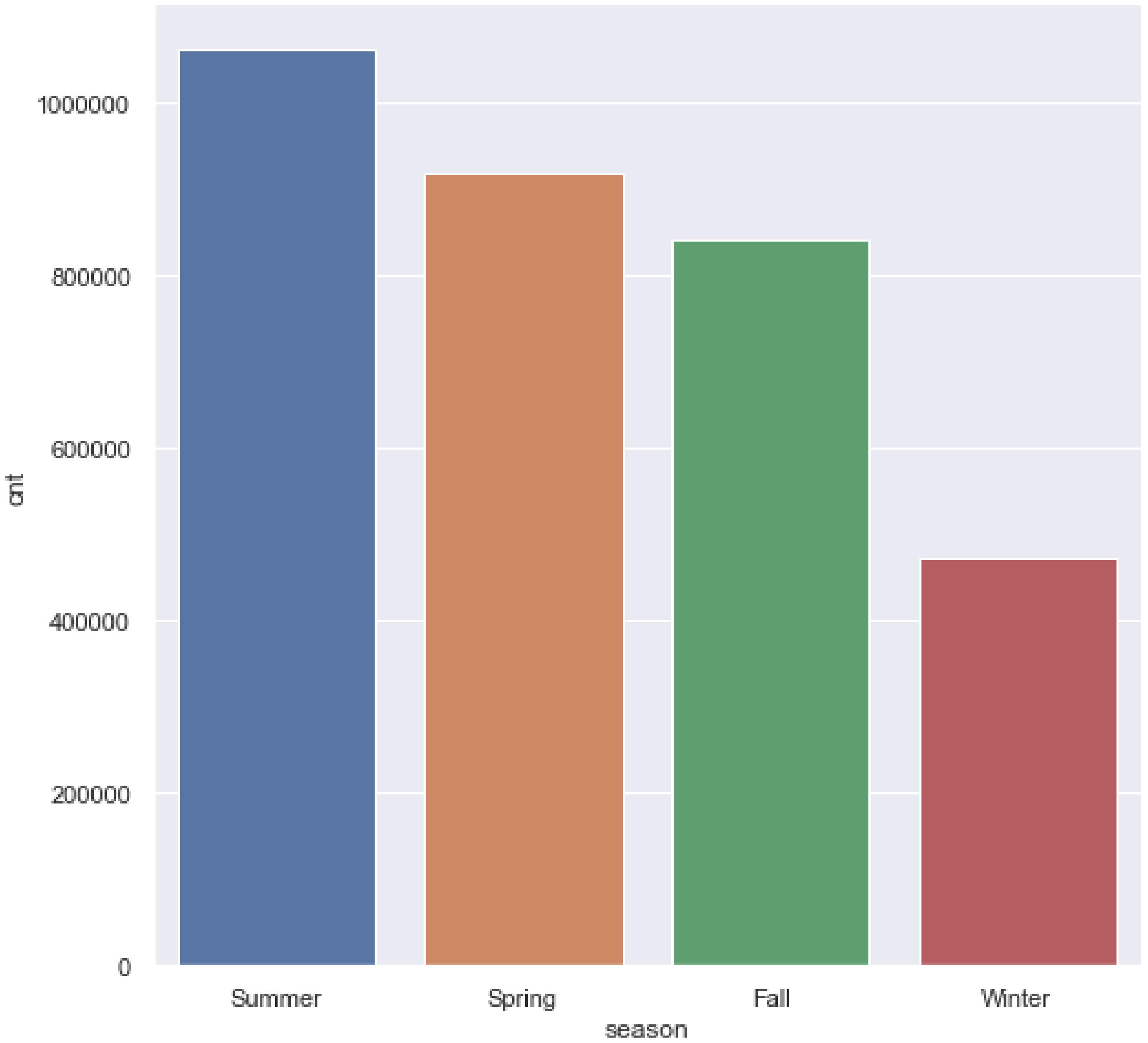

3000000
2500000
2000000
1500000
1000000
500000
0
cnt
0
1
holiday

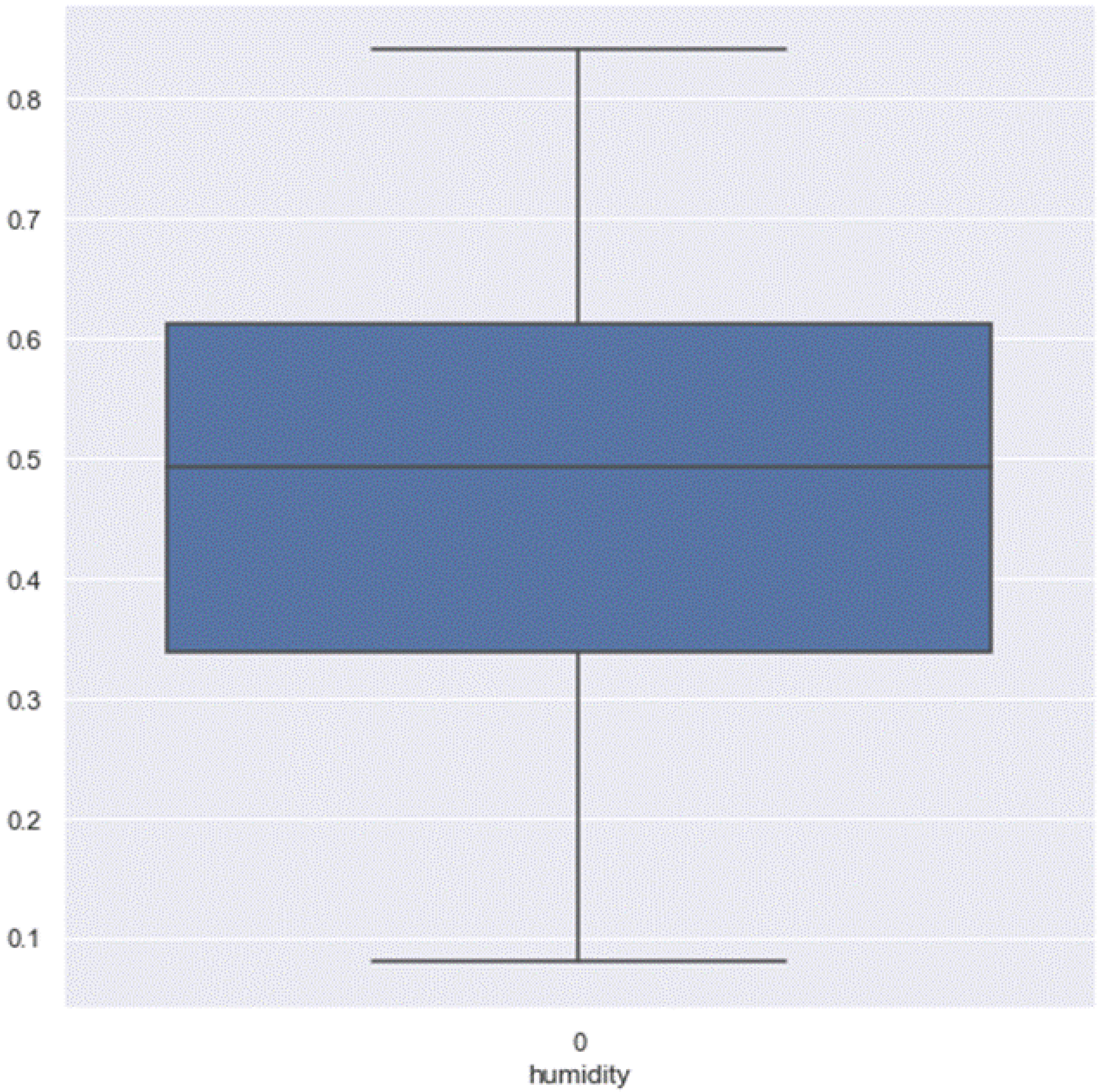


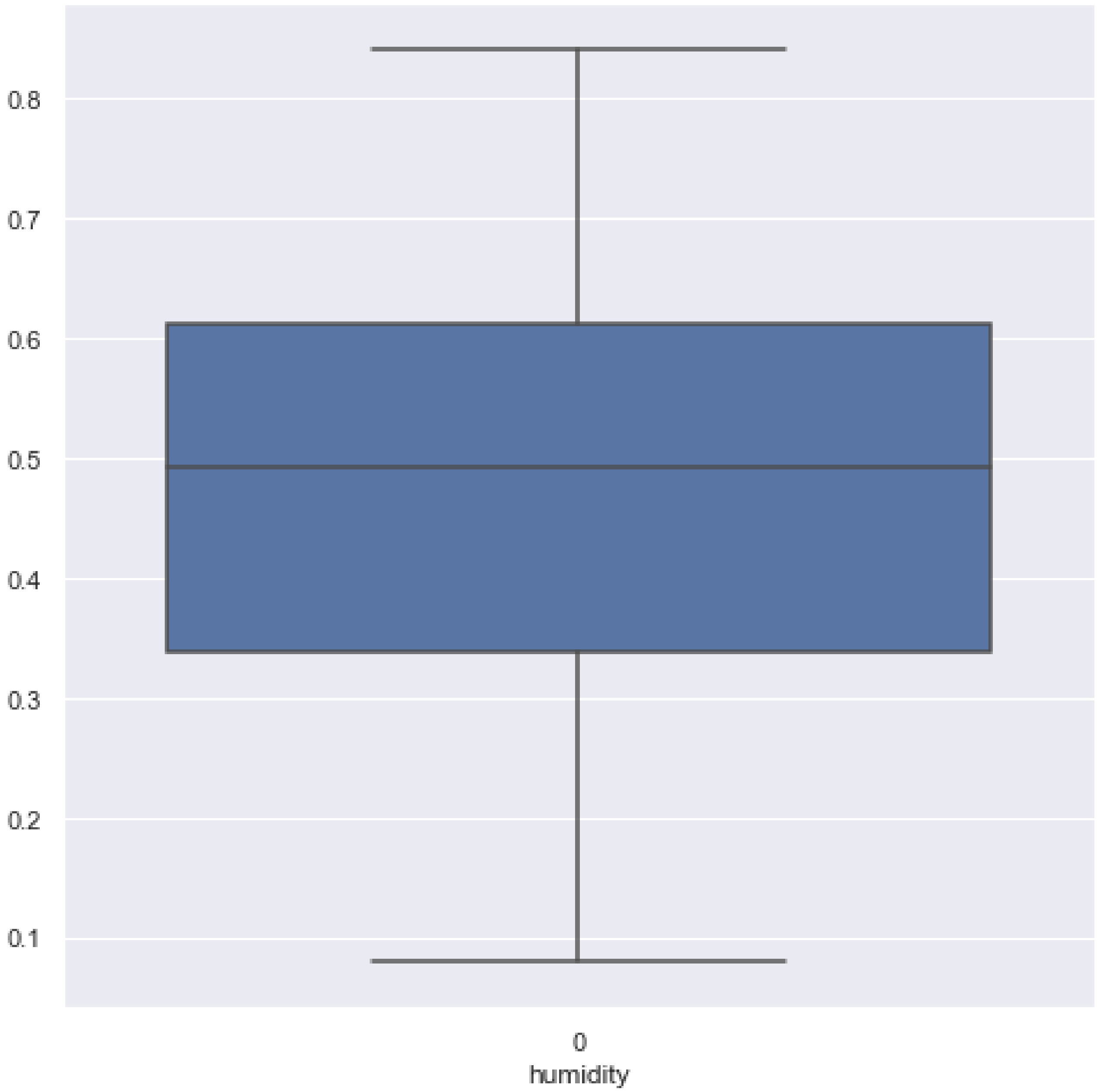
2000000
1500000
1000000
500000
0
cnt
Clear + Few clouds + Partly cloudy + Partly cloudy
Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
weathersit

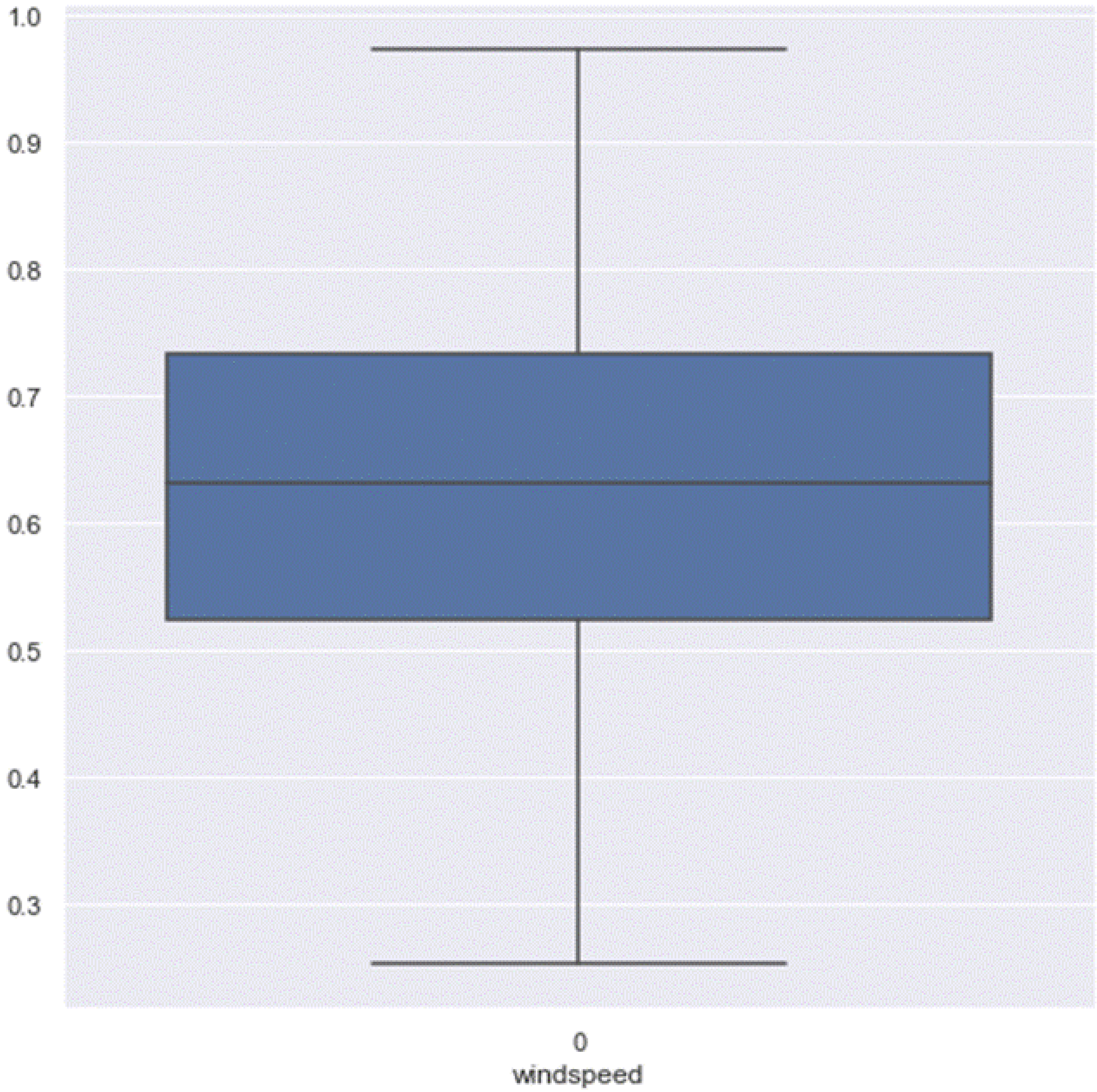

2000000
1500000
1000000
cnt

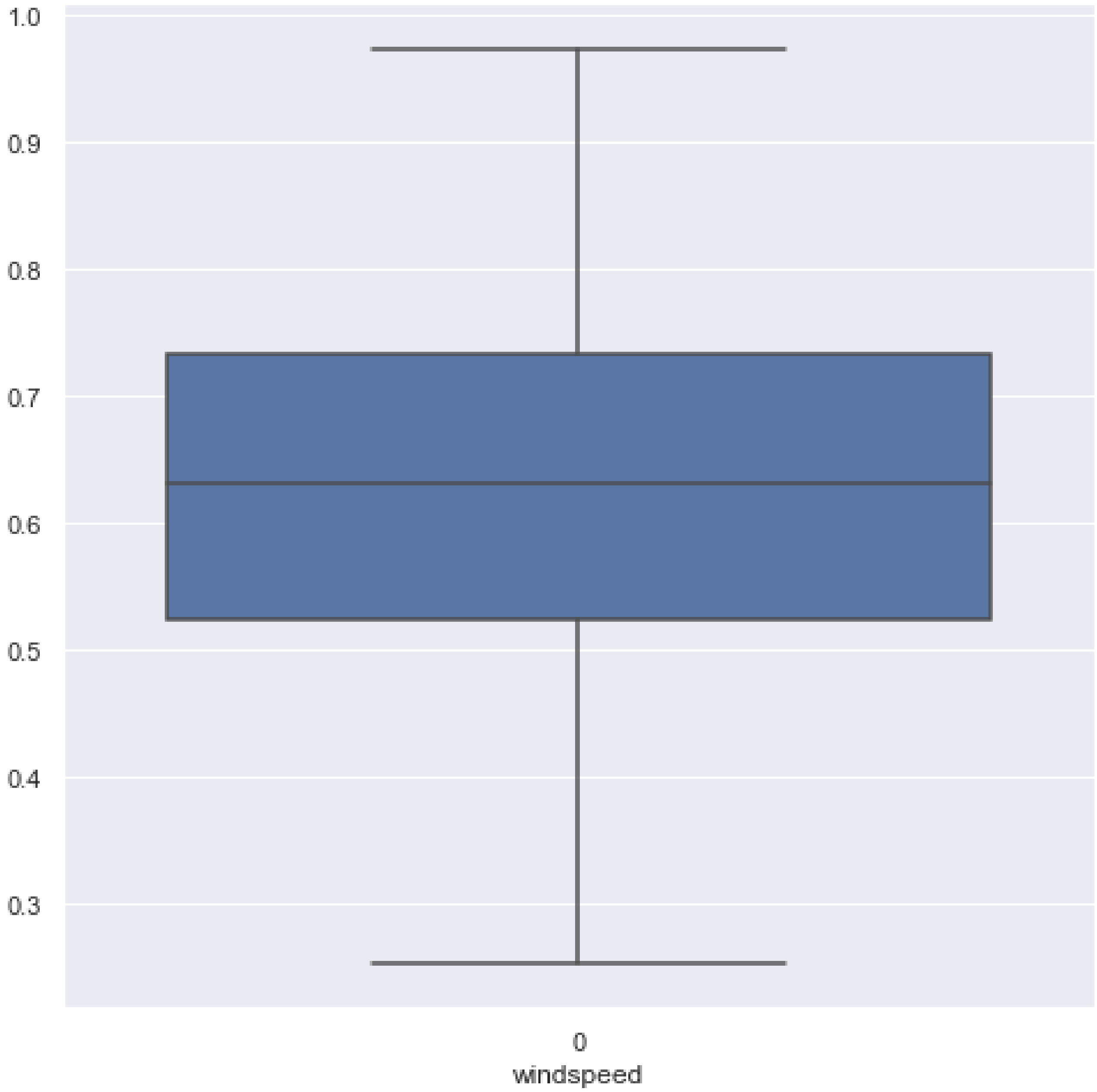

1000000
800000
600000
400000
200000
0
cnt
Summer
Spring
Fall
Winter
season

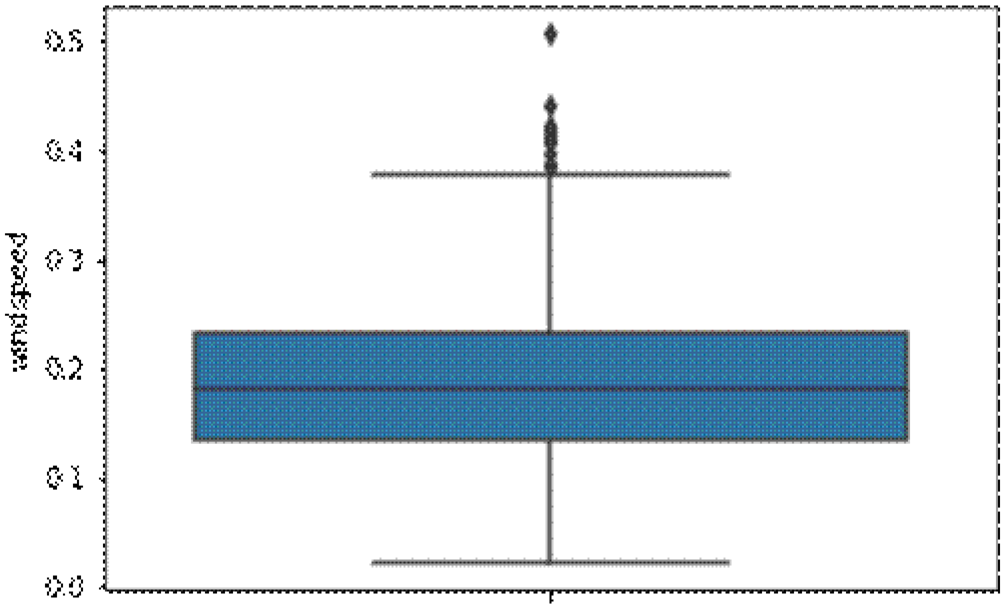

1000000
800000
600000
400000
200000
0
cnt
Summer
Spring
Fall
Winter
season

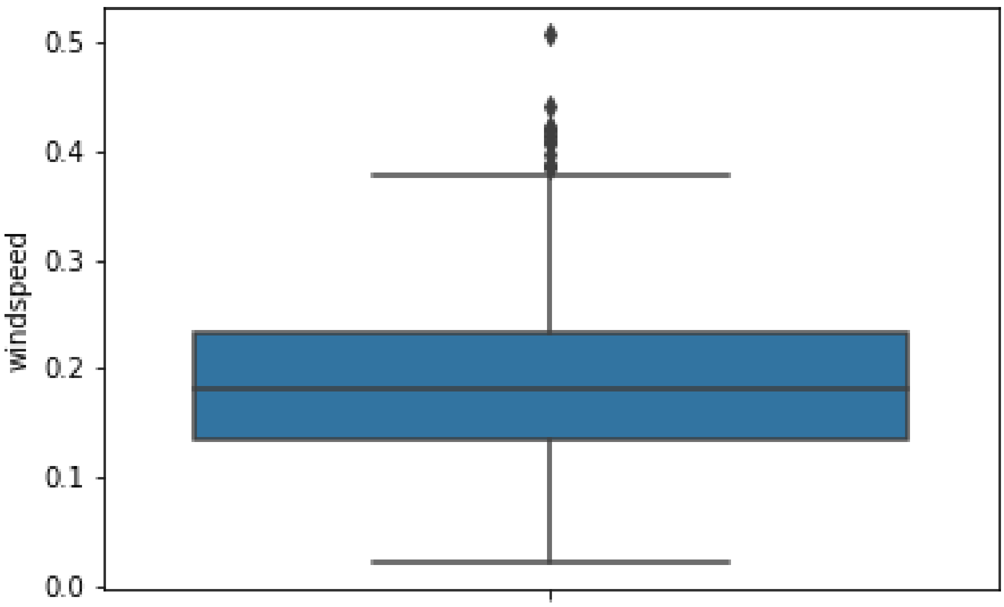

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
humidity

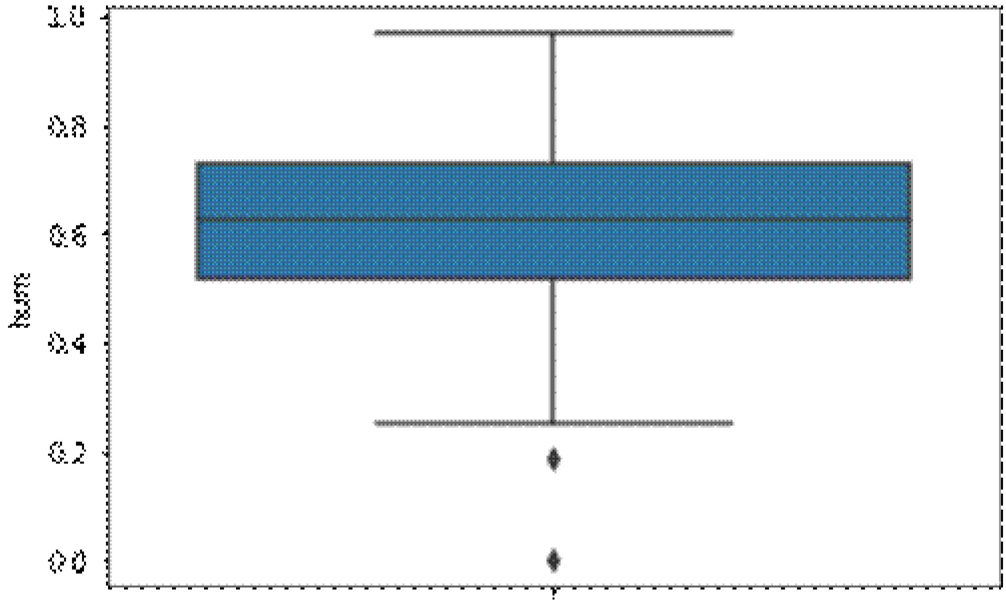
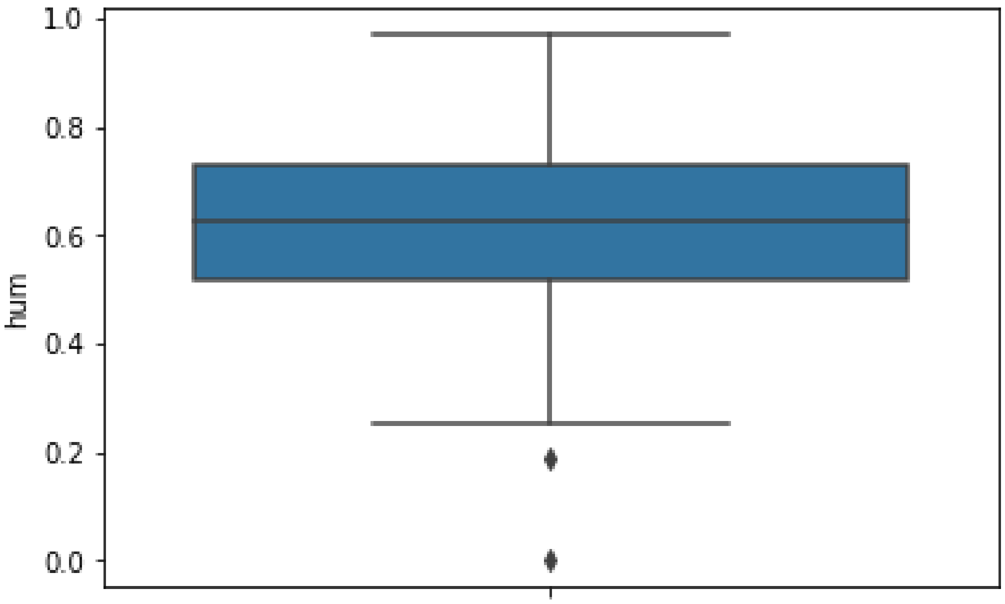

0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
humidity

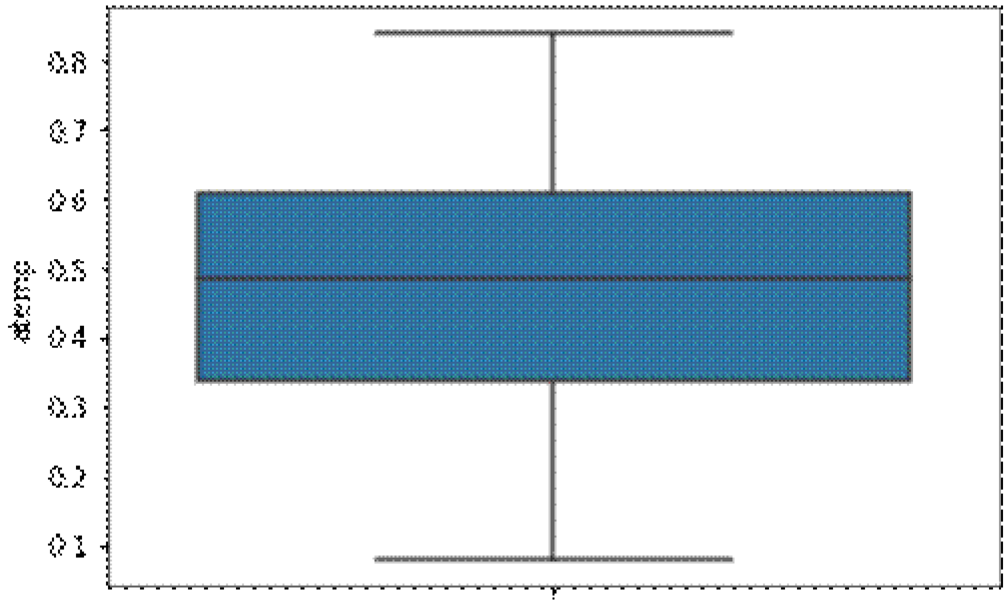

1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
windspeed


1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
windspeed

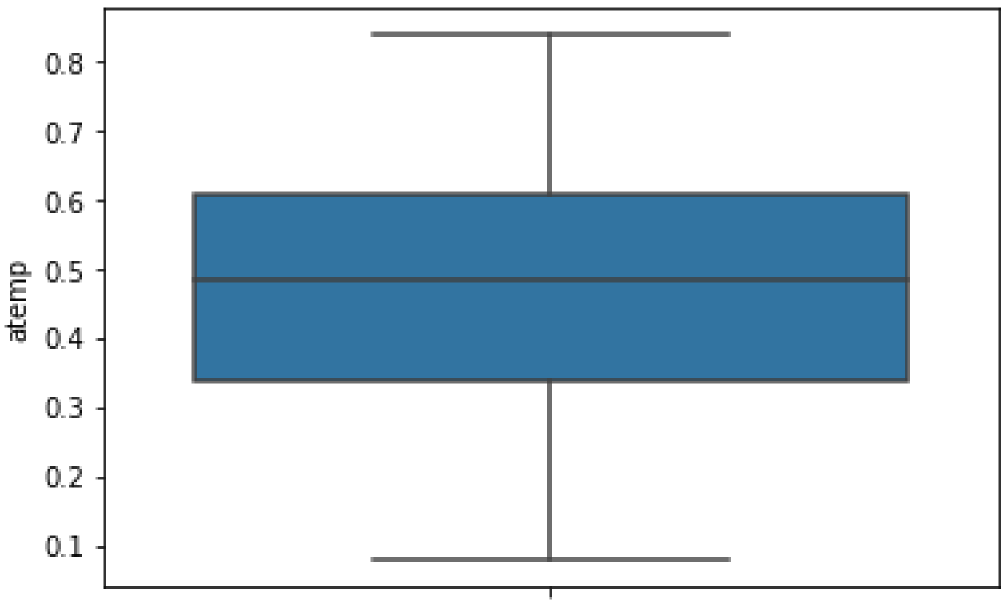

0.5
0.4
0.3
0.2
0.1
0.0
windspeed


0.5
0.4
0.3
0.2
0.1
0.0
windspeed

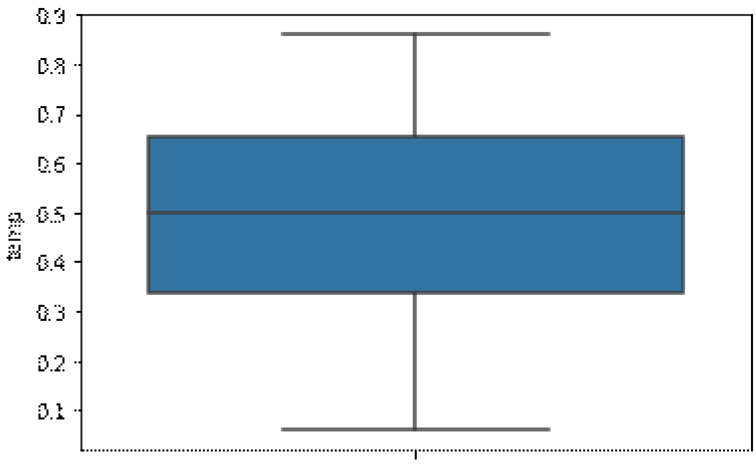

1.0
0.8
0.6
hum
0.4
0.2
0.0

1.0
0.8
0.6
hum
0.4
0.2
0.0



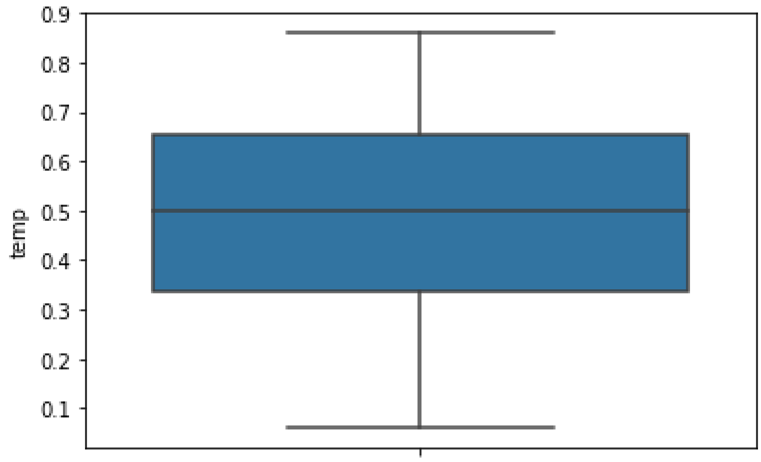
atemp
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

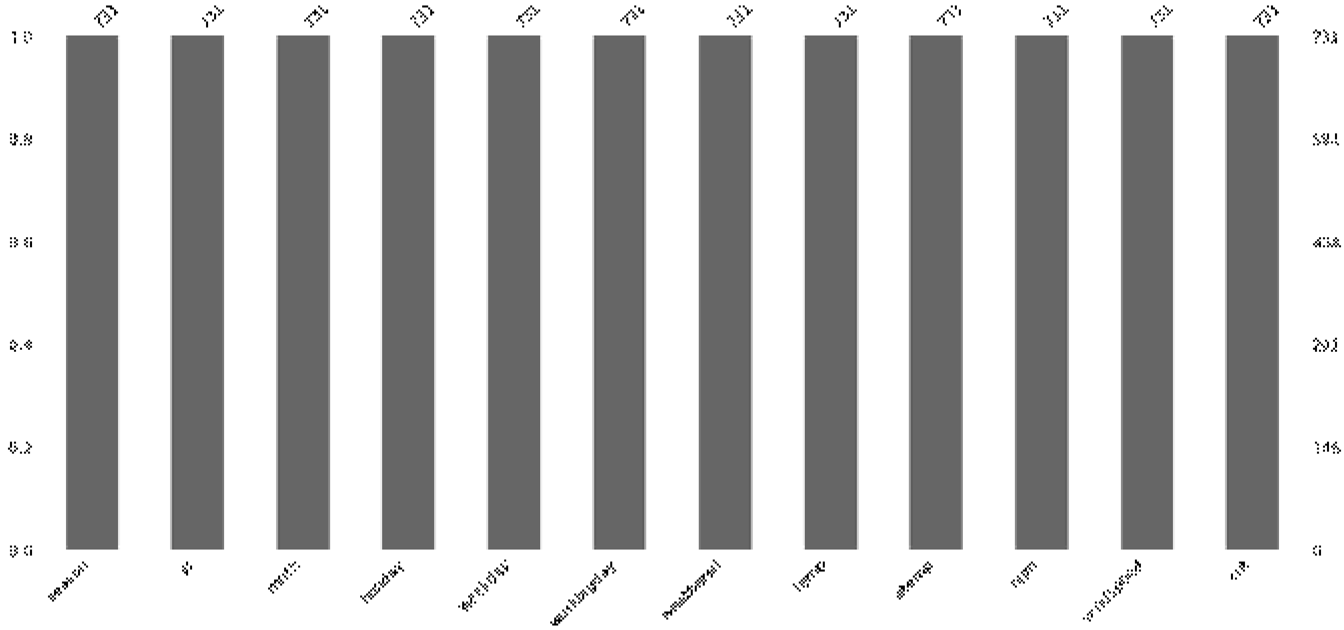


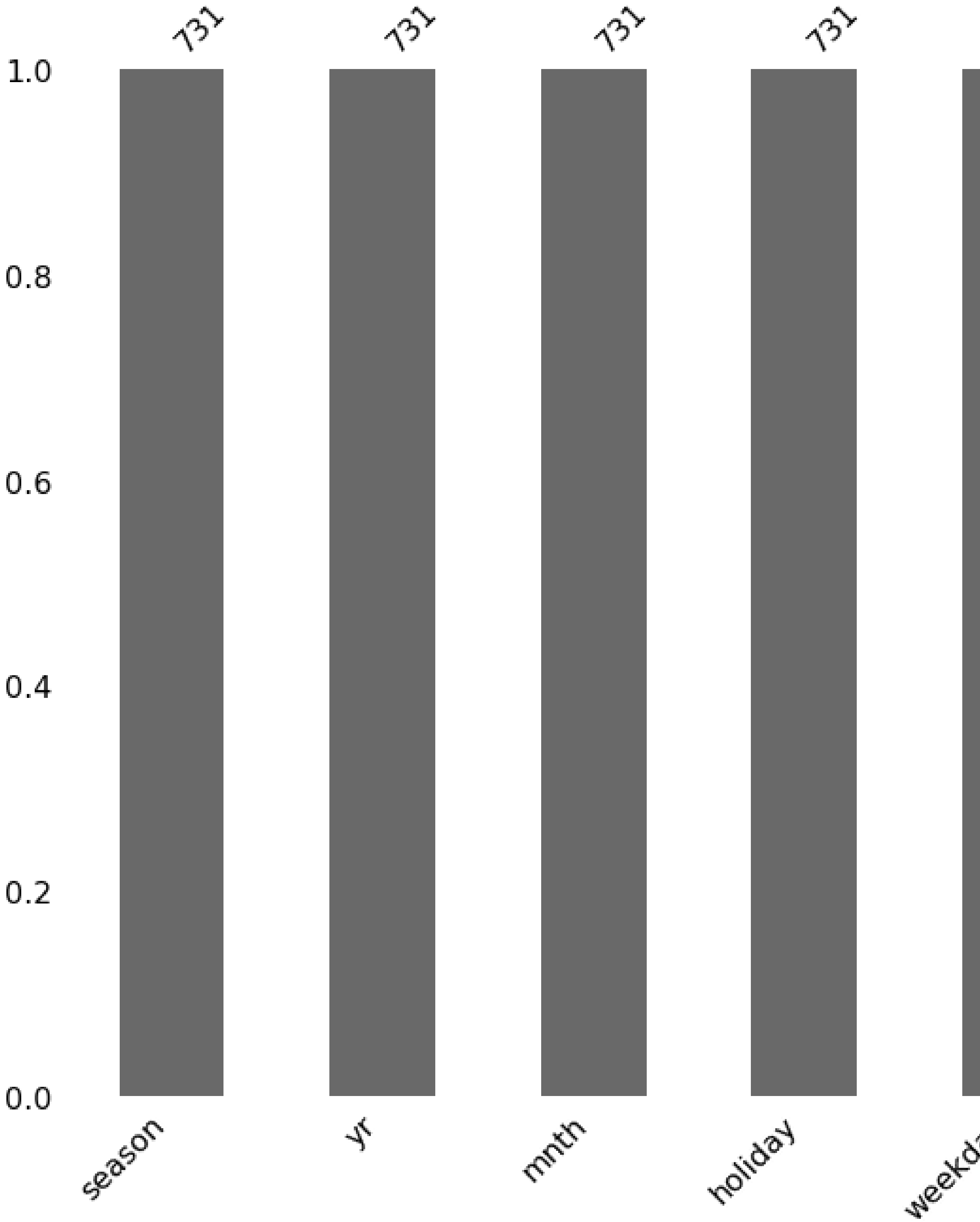
atemp
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1

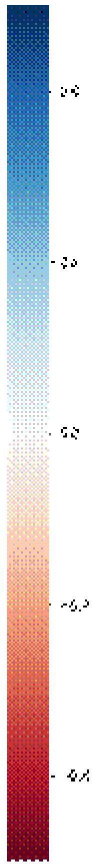


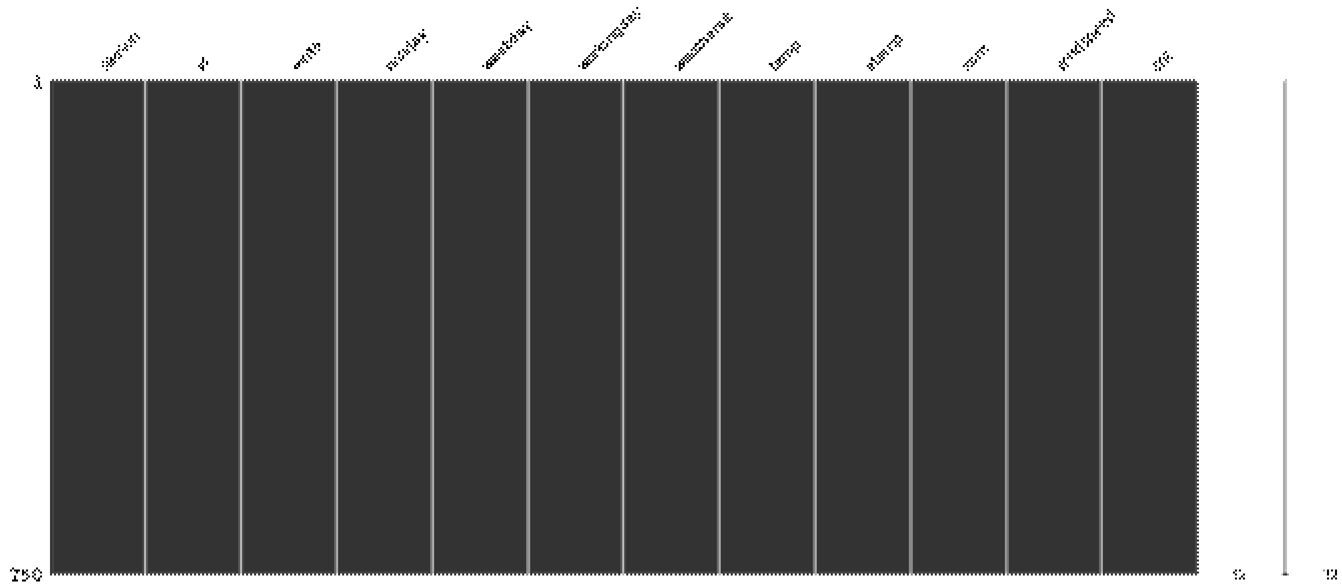
temp
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


temp
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1




731
731
731
731
1.0
0.8
0.6
0.4
0.2
0.0
season
yr
mnth
holiday





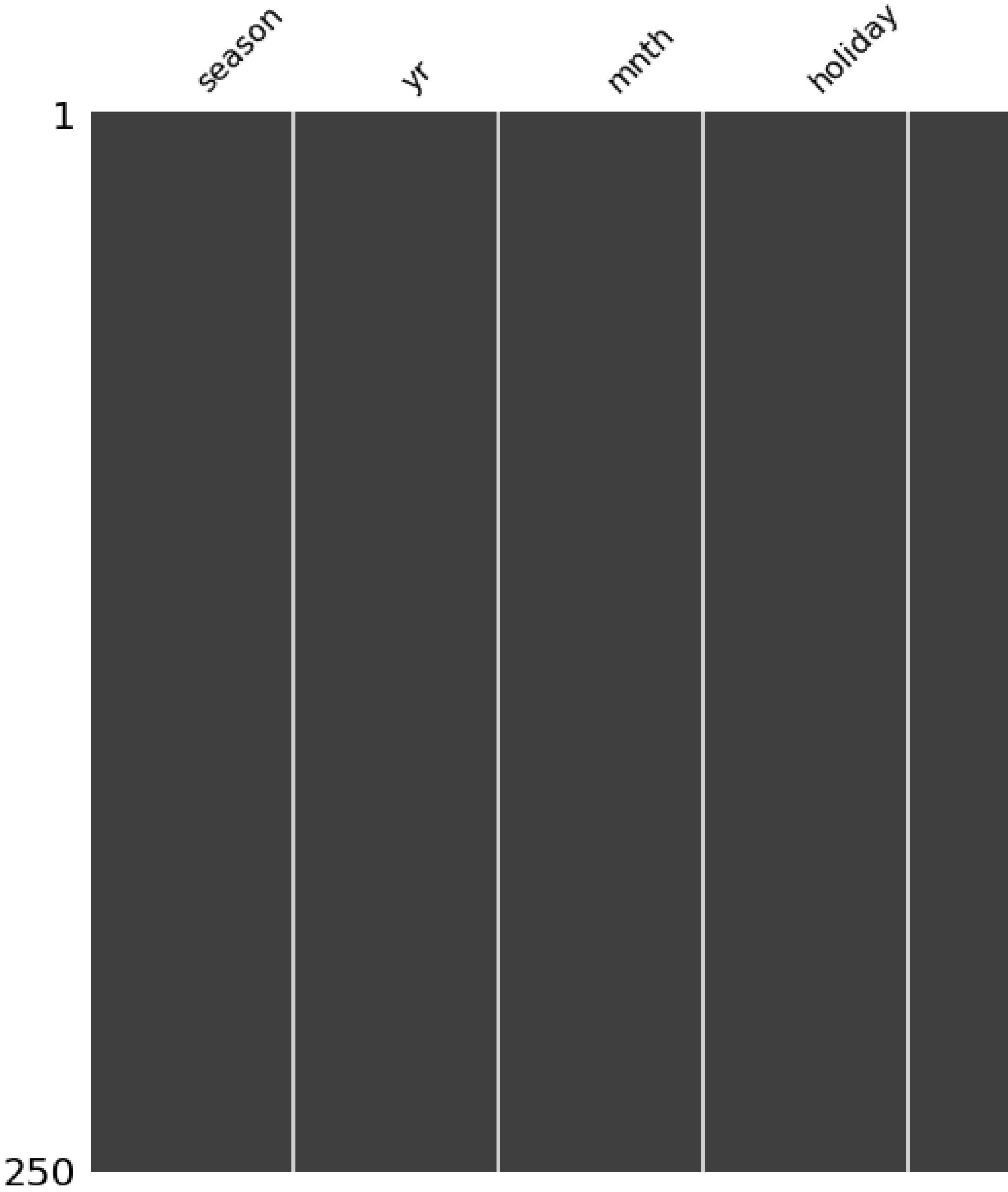

season
yr
mnth
holiday
1
250

Fig 2.7: Correlation plot of all the variables

Chapter 6: R code

```
setwd("D:\\edwisor\\project_bike _count")
day_bike=read.csv("day.csv")
summary(day_bike)
#checking data type of columns
sapply(day_bike, class)
#converting data type of dteday_bike into datetime
day_bike$weathersit = as.factor(day_bike$weathersit)
day_bike$season = as.factor(day_bike$season)
day_bike$dteday = as.character(day_bike$dteday)
day_bike$mnth = as.factor(day_bike$mnth)
day_bike$weekday = as.factor(as.character(day_bike$weekday))
day_bike$workingday = as.factor(as.character(day_bike$workingday))
day_bike$yr = as.factor(day_bike$yr)
day_bike$holiday = as.factor(day_bike$holiday)
#MISSING VALUE analysis
missing_val = sapply(day_bike, function(x){sum(is.na(x))})
#No missing values
#finding corelation value of numerical data
library(corrplot)
library(RColorBrewer)
data=Filter(is.numeric, day_bike)
corr <-cor(data)
corrplot(corr, type="upper", order="hclust",
    col=brewer.pal(n=8, name="RdYlBu"))
```

#it can be infered from curve as well as correlation matrix and plot that temp and atemp has strong relationship so oner should be dropped

#since casual and registered are to be predicted so we need to drop that also

day_bike=within(day_bike, rm(temp,casual,registered))

##dropping instant also as it is not needed

day_bike=within(day_bike, rm(instant))

target=subset(day_bike)

creating scatterplot of numerical variables with cnt

#install.packages('ggplot2')

library(ggplot2)

scat1 = ggplot(data = day_bike, aes(x =atemp, y = cnt)) + ggtitle("absolute Temperature") + geom_point() + xlab("absolute_Temperature") + ylab("Bike_Count")

scat2 = ggplot(data = day_bike, aes(x =hum, y = cnt)) + ggtitle(" Humidity") + geom_point(color="red") + xlab("Humidity") + ylab("Bike_Count")

scat3 = ggplot(data = day_bike, aes(x =windspeed, y = cnt)) + ggtitle(" Windspeed") + geom_point(color="red") + xlab("Windspeed") + ylab("Bike _Count")

gridExtra::grid.arrange(scat1,scat2,scat3,ncol=2)

#creating bar plot of categorical_variable

season_bar = ggplot(data = day_bike, aes(x = season)) + geom_bar() + ggtitle("Season_count")

weathersit_bar = ggplot(data = day_bike, aes(x = weathersit)) + geom_bar() + ggtitle(" Weather_count")

holiday_bar = ggplot(data = day_bike, aes(x = holiday)) + geom_bar() + ggtitle("Holiday_count")

workingday_bar = ggplot(data = day_bike, aes(x = workingday)) + geom_bar() + ggtitle("Working day_count")

Plotting plots together

gridExtra::grid.arrange(season_bar,weathersit_bar,holiday_bar,workingday_bar ,ncol=3)

###drawing boplot to get outliers.

```
boxplot(day_bike$hum)

boxplot(day_bike$windspeed)

boxplot(day_bike$atemp)

##outlier removal of hum and windspeed

outlier = day_bike[,'hum'][day_bike[,'hum'] %in% boxplot.stats(day_bike[,'hum'])$out]

day_bike = day_bike[which(!day_bike[,'hum'] %in% outlier),]

boxplot(day_bike$hum)

outlier = day_bike[,'windspeed'][day_bike[,'windspeed'] %in%
boxplot.stats(day_bike[,'windspeed'])$out]

day_bike = day_bike[which(!day_bike[,'windspeed'] %in% outlier),]

boxplot(day_bike$windspeed)

#checking feature importance

library(caret)

library(Boruta)

boruta = Boruta(cnt~., data = na.omit(day_bike), doTrace = 2)

# removing unwanted variables

day_bike=within(day_bike, rm(holiday,dteday))

#split data into train and test

#Divide the data into train and test

index = sample(1:nrow(day_bike), 0.75 * nrow(day_bike))

train_data = day_bike[index,]

test_data = day_bike[index,]



#random forest

randomforest_model = randomForest(cnt~., data = train_data, ntree = 500)


#Predict
```

```
preds = predict(randomforest_model, test_data[,-10])

#Create dataframe
df_mat = cbind(preds)
head(df_mat)

#Calculate MAPE
x1=regr.eval(trues = test_data[,10], preds = preds, stats = c("mae","mse","rmse","mape"))



########################################LINEAR REGRESSION########################################
#MAPE: 24%
#RMSE: 315.6434
#Accuracy:  0.8574
#MAE: 225.2275
#Adjusted R squared: 0.8498
#F-statistic: 112.9

#Train the data using linear regression
linear_regression = lm(formula = cnt~., data = train_data)

#summary of the model
summary(linear_regression)

#Predict
predictions = predict(linear_regression, test_data[,-10])
```

```
#Create dataframe

df_mat = cbind(df_mat,predictions)

head(df)


#MAPE

x2=x1=regr.eval(trues = test_data[,10], preds = preds, stats = c("mae","mse","rmse","mape"))
```