Bike Sharing Data Analysis for Casual & Registered Users in 2011 - 2012

Created by Eka Pramudianzah

Study Case Background

A traditional bicycle rental services company, “GowesKuyy” provides a bike sharing services where the process of membership, rental, and returning the bicycle has become automatically. With automatic system, customers/users can easily rent a bike from a certain location and return back at another location

Currently with the huge market interest in the automated system created by the “GowesKuyy” Company, there are already 500 bike sharing programs around the world covering more than 500 bicycles

Analysis Questions

1. What is the comparison between the total number of bicycle users in 2011 & 2012, both registered and casual customers in each season?

2. What is the level of comparison of the frequency/number of data values based on the percentage of bicycle users on working days and weekend/holiday?

3. How is the trend towards bicycle use by both types of customers (registered & casual customers) during the loan period from January 1, 2011 - December 31, 2012 based on working days and weekend/holiday?

4. What is the average number of bicycle users each day?

5. What are the trends in bicycle use from registered and casual users based on date of use in each weather category?

Table of contents

01

Python Data Analysis Libraries

Libraries Used

pandas

Pandas

NumPy

Numerical Python (NumPy)

matplotlib

Matplotlib

seaborn

Seaborn

Install the Python Libraries

If we don’t have the Python libraries such as Pandas, NumPy, etc., we can install in Notebook cell (if we use Jupyter Notebook) and Command Prompt with scripts:

1. **Installing Pandas Library:**
   pip install pandas

2. **Installing NumPy Library:**
   pip install numpy

3. **Installing Matplotlib Library:**
   pip install matplotlib

4. **Installing Seaborn Library:**
   pip install seaborn

```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\ekapr\anaconda3\lib\site-packages (1.4.2)
Requirement already satisfied: numpy>=1.18.5 in c:\users\ekapr\anaconda3\lib\site-packages (from pandas) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in c:\users\ekapr\anaconda3\lib\site-packages (from pandas) (2021.3)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\ekapr\anaconda3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\ekapr\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas) (1.
16.0)
```

How to Use Python Libraries

Import pandas **as** pd
Import numpy **as** np
Import matplotlib.pyplot **as** plt
Import seaborn **as** sns
plt.style.use(“ggplot”)

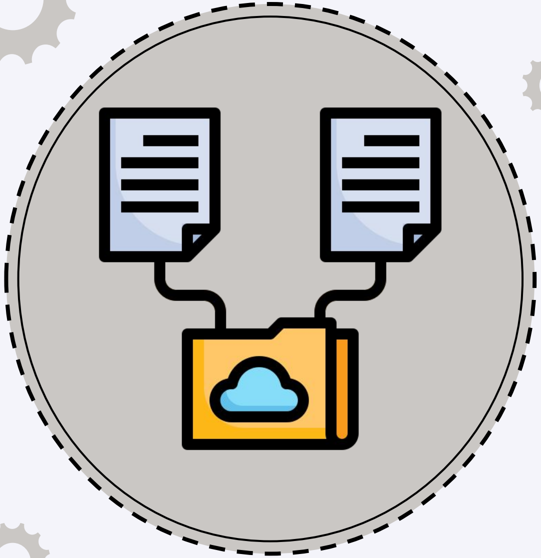
An additional note, the **“as”** function in the import section aims to provide an alias for each library that will be use in Data Processing, especially in Python

```
import matplotlib.pyplot as plt
from matplotlib import style

print(plt.style.available)
```

```
['Solarize_Light2', '_classic_test_patch', '_mpl-gallery', '_mpl-gallery-nogrid', 'bmh', 'classic', 'dark_background', 'fast',
'fivethirtyeight', 'ggplot', 'grayscale', 'seaborn', 'seaborn-bright', 'seaborn-colorblind', 'seaborn-dark', 'seaborn-dark-pale
tte', 'seaborn-darkgrid', 'seaborn-deep', 'seaborn-muted', 'seaborn-notebook', 'seaborn-paper', 'seaborn-pastel', 'seaborn-post
er', 'seaborn-talk', 'seaborn-ticks', 'seaborn-white', 'seaborn-whitegrid', 'tableau-colorblind10']
```

Apart from that, the purpose of using **"plt.style.use("ggplot")”** is to provide new variations such as color, graph background, line thickness, etc., to visualization graphs from the Matplotlib & Seaborn Libraries



02

Collect and Assessing Dataset

Dataset Source and Description

Bike Sharing Dataset has been successfully downloaded from the UCI Machine Learning Repository, which contains a database collection system, domain theories, and data generator used by the machine learning community for empirical analysis of Machine Learning Algorithms

day.csv

hour.csv

Bike Sharing Dataset have 2 files of Comma Separated Values (.csv). But in this portfolio, I just use day.csv file only because the main idea is to analyze the use of rental bicycles in each day

In the dataset file "day.csv" there are 16 column attributes, while in the dataset file "hour.csv" there are 17 column attributes The difference is the "hr" attribute is only in the "hour.csv" dataset file. Meanwhile the "day.csv" dataset file doesn’t have that attribute

Read Dataset File using Pandas

One of the script code for load/read a dataset file into a DataFrame using Pandas Library:

```
df_bikesharing_per_day = pd.read_csv("Data/day.csv")
```

As a note, you need to know that the Bike Sharing Dataset File “day.csv” has been placed in a sub-local folder so the way to load the dataset file is like:

df_bikesharing_per_day = pd.read_csv(“Data/day.csv”)

1. **“pd”** is alias from Pandas Library
2. **df_bikesharing_per_day** is a DataFrame variable which contains the data in the "day.csv" file. When we want to analyze data in Python, we have to create a DataFrame Variable and there is no definite rule for naming a DataFrame variable (but most likely of Analysts give the variable name as "df")
3. **"read_csv”** is a function from the Pandas Library to read/load dataset files that have CSV/Comma Separated Values format
4. **“Data”** is a sub-local folder
5. **day.csv** is a dataset file use in this portfolio

Description of Each Columns in the DataFrame - Part 1

Variable Name	Data Type	Description
instant	integer	Rows index of each data. The function of “instant” column as a primary key of the Bike Sharing Dataset Table
dteday	object	The "dteday" column explain the period of bicycle use by customers from January 1 2011 - December 31 2012
season	integer	Describe seasons based on numerical values (1. Spring, 2. Summer, 3. Fall/Autumn, 4. Winter)
yr	integer	The “yr” column explain the year rental of bike sharing (0: 2011, 1: 2012)

Description of Each Columns in the DataFrame - Part 2

Variable Name	Data Type	Description
mnth	integer	“mnth” column explain the period of bicycle use in monthly (1 to 12)
holiday	integer	Explain whether the day/date is a holiday or not (0: Not Holiday, 1: Holiday)
weekday	integer	Day of the week (6: Saturday, 0: Sunday, 1: Monday, 2: Tuesday, 3: Wednesday, 4: Thursday, 5: Friday)
workingday	integer	The “workingday” column explain the status of working day (0: weekend/holiday, 1: working day)

Description of Each Columns in the DataFrame - Part 3

Variable Name	Data Type	Description
weathersit	integer	This column explains the weather status written in numerical numbers (1: Clear Cloudy, 2: Mist Cloudy, 3: Snow + Light Rain + Thunderstorm, 4: Heavy Rain + Mist + Thunderstorm)
temp	float	Normalized temperature in Celcius
atemp	float	Normalized average temperature in Celcius
hum	float	The “hum” column explain the normalized humidity with numerical decimal

Description of Each Columns in the DataFrame - Part 4

Variable Name	Data Type	Description
windspeed	float	Normalized wind speed with numerical decimal
casual	integer	“casual” column explain the count of casual users where casual users is users not registered in automatic system from “GowesKuyy” Company
registered	integer	“registered” column explain the count of registered users where registered users is user are registered in automatic system from “GowesKuyy” Company
cnt	integer	Count of total users including both casual and registered users

Assessing Data

Displays Some Important Information on the DataFrame

```
df_bikesharing_per_day.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   instant     731 non-null    int64
 1   dteday      731 non-null    object
 2   season      731 non-null    int64
 3   yr          731 non-null    int64
 4   mnth        731 non-null    int64
 5   holiday     731 non-null    int64
 6   weekday     731 non-null    int64
 7   workingday  731 non-null    int64
 8   weathersit  731 non-null    int64
 9   temp        731 non-null    float64
 10  atemp       731 non-null    float64
 11  hum         731 non-null    float64
 12  windspeed   731 non-null    float64
 13  casual      731 non-null    int64
 14  registered  731 non-null    int64
 15  cnt         731 non-null    int64
dtypes: float64(4), int64(11), object(1)
memory usage: 91.5+ KB
```

Displays the Amount of Duplicated Data in a DataFrame

```
df_bikesharing_per_day.duplicated().sum()
```

```
0
```

Displays the Number of Rows & Columns

```
df_bikesharing_per_day.shape
```

```
(731, 16)
```

Displays the Number of Missing Values from each Attributes

```
df_bikesharing_per_day.isnull().sum()
```

```
instant       0
dteday        0
season        0
yr            0
mnth          0
holiday       0
weekday       0
workingday    0
weathersit    0
temp          0
atemp         0
hum           0
windspeed     0
casual        0
registered    0
cnt           0
dtype: int64
```

03

Data Preprocessing (Data Cleaning & Manipulation)

Introduction to Data Preprocessing

	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

Because the dataset currently used doesn’t have duplicated data or missing values, so in the Data Cleaning,I will carry out a change/transformation for values identified as invalid values

Several attributes that are known to have a value format that doesn’t comply with the background of the explanation attributes in the UCI Machine Learning Repository include the "season", "yr", “holiday”, “weekday”, "workingday" and "weathersit" attributes

Data Preprocessing Steps

01

Remove Unnecessary Attributes

02

Renaming Columns

03

Inappropriate Data Value
Format Transformation

04

Change the Data Types &
Create New Attribute

Remove Unnecessary Attribute

Top 5 rows before “instant” attributes are removed

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

Top 5 rows after “instant” attributes are removed

	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

Code script for remove “instant” attribute in DataFrame list columns:

df_bikesharing_per_day.drop([“instant”], axis=1)

Deleting the "instant" column from the list of columns in the DataFrame is intended because the existence of the "instant" column is already represented by an index in the DataFrame

Renaming Columns

List of columns before transformation

```
Index(['dteday', 'season', 'yr', 'mnth', 'holiday', 'weekday', 'workingday',
       'weathersit', 'temp', 'atemp', 'hum', 'windspeed', 'casual',
       'registered', 'cnt'],
      dtype='object')
```

List of columns after transformation

```
Index(['Rental_Date', 'Season', 'Year', 'Month', 'Holiday', 'Weekday',
       'Working_Day', 'Weather', 'Temperature', 'Average_Temperature',
       'Humidity', 'Windspeed', 'Casual', 'Registered', 'Total'],
      dtype='object')
```

There are 2 ways in this portfolio to renaming columns with different code scripts, including:

1. df_bikesharing_per_day.rename(columns ={"dteday": "rental_date", "yr": "year", "mnth": "month", "workingday": "Working_Day", "weathersit": "weather", "temp": "temperature", "atemp": "average_temperature", "hum": "humidity", "cnt": "total"})

2. df_bikesharing_per_day.rename(str.title, axis=1)

Inappropriate Data Value Format Transformation

Several columns that will be transformed into an appropriate value format based on the description of the column include:

1. Season
2. Year
3. Holiday
4. Weekday
5. Working_Day
6. Weather

This transformation must be carried out because if the data values are only written in numerical numbers even though the data is categorical data, it will make it difficult to interpret and draw conclusions about the data

One of the example about code script is:
df_bikesharing_per_day[“Year”].**replace**([0, 1], [2011, 2012])

```
3    188
2    184
1    181
4    178
Name: Season, dtype: int64

1    366
0    365
Name: Year, dtype: int64

0    710
1     21
Name: Holiday, dtype: int64

1    500
0    231
Name: Working_Day, dtype: int64

1    463
2    247
3     21
Name: Weather, dtype: int64

6    105
0    105
1    105
2    104
3    104
4    104
5    104
Name: Weekday, dtype: int64
```

Transformation

```
Fall      188
Summer    184
Spring    181
Winter    178
Name: Season, dtype: int64

2012    366
2011    365
Name: Year, dtype: int64

Not Holiday    710
Holiday         21
Name: Holiday, dtype: int64

Working Day        500
Weekend/Holiday    231
Name: Working_Day, dtype: int64

Clear Cloudy                            463
Mist Cloudy                             247
Snow + Light Rain + Thunderstorm         21
Name: Weather, dtype: int64

Saturday     105
Sunday       105
Monday       105
Tuesday      104
Wednesday    104
Thursday     104
Friday       104
Name: Weekday, dtype: int64
```

Change the Data Types & Create New Attribute

```
<class 'pandas.core.series.Series'>
RangeIndex: 731 entries, 0 to 730
Series name: Rental_Date
Non-Null Count  Dtype
--------------  -----
731 non-null    object
dtypes: object(1)
memory usage: 5.8+ KB
```

```
<class 'pandas.core.series.Series'>
RangeIndex: 731 entries, 0 to 730
Series name: Rental_Date
Non-Null Count  Dtype
--------------  -----
731 non-null    datetime64[ns]
dtypes: datetime64[ns](1)
memory usage: 5.8 KB
```

Change Format and Data Type for "Rental_Date" Column

df_bikesharing_per_day[“Rental_Date”] = **pd.to_datetime**(df_bikesharing_per_day[“Rental_Date”])

```
0    1
1    1
2    1
3    1
4    1
Name: Month, dtype: int64
```

```
0    January
1    January
2    January
3    January
4    January
Name: Month, dtype: object
```

Change Format and Data Type for "Month" Column

df_bikesharing_per_day[“Month”] = **pd.to_datetime**(df_bikesharing_per_day["Month"], **format="%m"**).**dt.strftime("%B")**

```
Index(['Rental_Date', 'Season', 'Year', 'Month', 'Holiday', 'Weekday',
       'Working_Day', 'Weather', 'Temperature', 'Average_Temperature',
       'Humidity', 'Windspeed', 'Casual', 'Registered', 'Total', 'Rental_Day'],
      dtype='object')
```

Create a New Attribute Variable, “Rental_Day”

df_bikesharing_per_day**["Rental_Day"]** = df_bikesharing_per_day**["Rental_Date"].dt.day**

04

Exploratory Data Analysis (EDA)

Explore the Most Frequency of Data Values from the "Season" Attribute

```
df_bikesharing_per_day["Season"].value_counts().sort_values(ascending=False)
```

```
Fall      188
Summer    184
Spring    181
Winter    178
Name: Season, dtype: int64
```

The results below show that the season that appears most frequently in the "Season" column is autumn/fall with a total of 188 values

I think this makes perfect sense because autumn/fall season is a very suitable season for cycling. Apart from that, another reason why it is autumn (fall season) in my opinion is because autumn occurs after summer (summer season) and in my opinion this is a habit for the majority of European society

Explore Total Number, Standard Deviation, and Average Bike Users by Year of Rental

```
df_bikesharing_per_day.groupby("Year").agg({"Casual": ["sum", "mean", "std"],
                                            "Registered": ["sum", "mean", "std"]})
```

	Casual			Registered		
	sum	mean	std	sum	mean	std
Year						
2011	247252	677.402740	556.269121	995851	2728.358904	1060.110413
2012	372765	1018.483607	758.989897	1676811	4581.450820	1424.331846

The results above show that in 2011 the number of casual bicycle users reached 247,252. Then in 2012, the number of users increased with a total value reaching 372,765

Then in 2011 with the registered user type, the number of users increased almost 2x compared to the number of casual users in the same year (2011) with a total value reaching 995,851. Meanwhile in 2012, the number of registered users reached 1,676,811 users

In my opinion, this trend is very good from a business perspective, in 2012 we can improvise in terms of business, especially in increasing the number of users

Explore the Total Users based on Rental Months and Seasons

```
pivot_table_by_month_and_season = pd.pivot_table(df_bikesharing_per_day, values="Total",
                                                 index=["Month", "Season"], aggfunc="sum")

sorted_pivot_table = pivot_table_by_month_and_season.sort_values(by=["Total"], ascending=False)

sorted_pivot_table
```

Month	Season	Total
August	Fall	351194
July	Fall	344948
May	Summer	331686
October	Winter	322352
April	Summer	269094
November	Winter	254831
September	Fall	249599
June	Summer	230954
December	Winter	168038
February	Spring	151352
March	Spring	142065
January	Spring	134933
June	Fall	115388
September	Winter	96392
March	Summer	86855
December	Spring	42998

The results also show that the third month (August) has the highest number of users compared to other Autumn/Fall months such as June, July and September

Apart from that, the number of users in August also had the highest number of users compared to other months with a total value of 351.194

On the other hand, Spring season, which occurs in December have the lowest number of bicycle users among the other months just only 42.998

This shows that customers are not interested in cycling in December during the Spring season so the number of rental bicycle users is very low

Explore Total Number of Bicycle Users on Holidays & Working Days in each Loan Year (2011 & 2012)

```
pivot_table_bikesharing_per_year = pd.pivot_table(df_bikesharing_per_day, values="Total", index=["Year", "Working_Day"],
                                                   aggfunc="sum")

reset_pivot_table_index = pivot_table_bikesharing_per_year.reset_index().sort_values(
    ["Year", "Total"], ascending=[True, False]).set_index(["Year", "Working_Day"])

reset_pivot_table_index
```

Year	Working_Day	Total
2011	Working Day	856264
	Weekend/Holiday	386839
2012	Working Day	1436146
	Weekend/Holiday	613430

Based on these results, I can assume that the background of the majority of bicycle users comes from workers/students because the highest total value of using borrowed bicycles occurs during work days or school days

Meanwhile, on weekends or holidays, which in fact are relaxing days and are very suitable for cycling activities, it turns out that in reality the number of bicycle users on weekends is not high enough and does not even exceed the number of bicycle users on weekdays

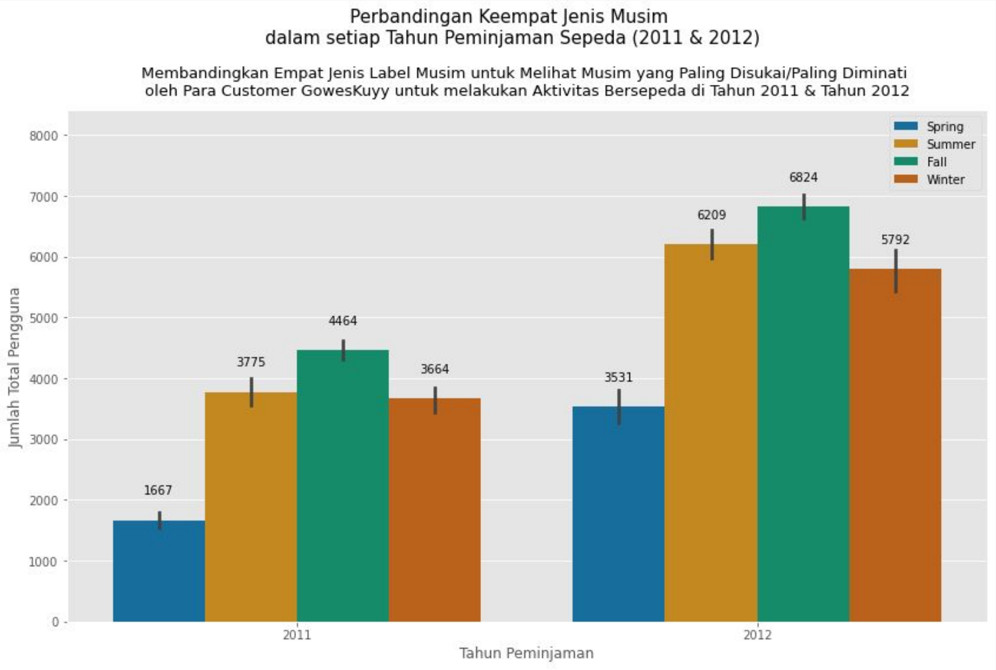
In 2011, the total number of bicycle users apparently came from working days with a total of 856,264 users, while the number of users who borrowed bicycles on holidays/weekends only amounted 386,839

Then in 2012 the highest total number of bicycle users on working days is 1,436,146 users, while the total number of bicycle users who borrowed bicycles on holidays was only 613,430

05

Data Visualization with Analysis

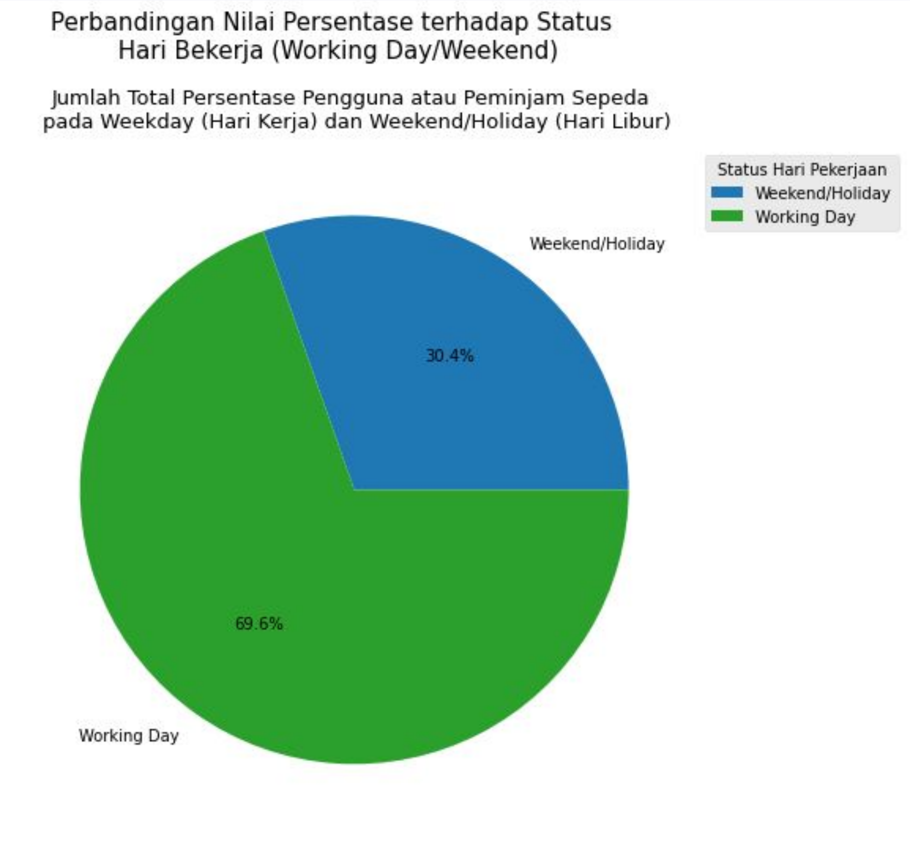
First Question: What is the Comparison between the Total Number of Bicycle Users in 2011 & 2012, both Registered and Casual Customers in Each Season?



The besides bar chart shows that if calculated as a whole between each total number of bicycle users owned by the four season labels in each loan year, then 2012 is the year that has the highest & highest total number of borrowers compared to 2011. But of the two loan years There is one similarity, namely fall/autumn season is the season that has the highest total number of bicycle users among the other seasons

The number of bicycle users during the fall of 2011 was 4,464 people, while the number of bicycle users who used bicycle rental services during the fall of 2012 was 6,824 people. This means that there is a trend of increasing the number of new users because they managed to get as many as 2,360 users

Second Question: How the Comparison about Frequency/Number of Data Values based on the Percentage of Bicycle Users on Working Days and Weekend?

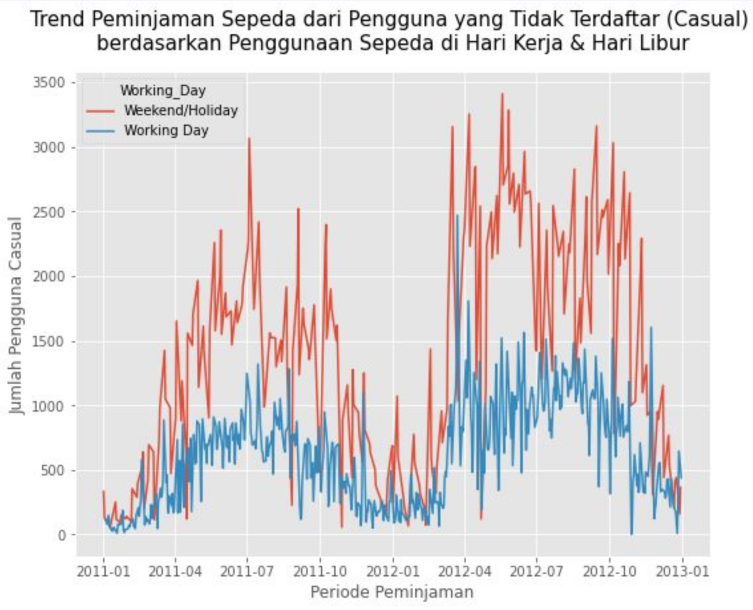


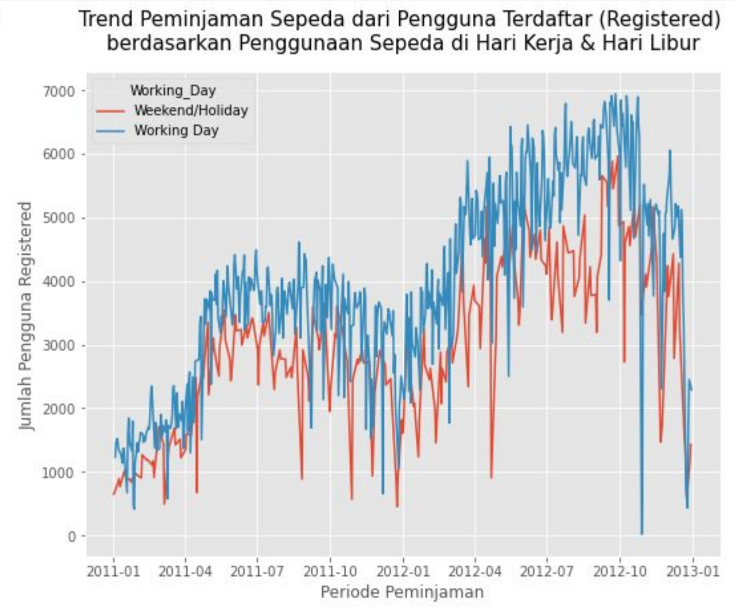
Based on the pie chart on the side, it shows that employees or even school students and university students always ride bicycles

The number of rental bicycle users on working days is very large with a percentage of 69.6% compared to rental bicycle users on holidays and weekends (30.4%)

This means that the majority of bicycle users come from employees/school students as well as university students because the majority of their activities are carried out on Monday - Friday, which are working days

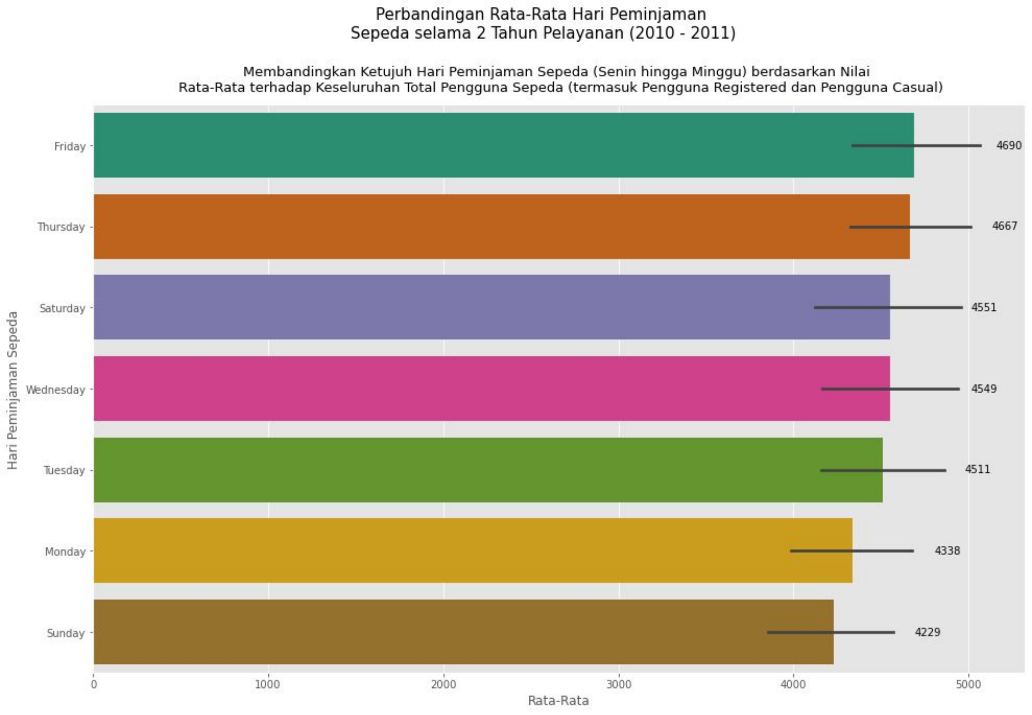
Third Question: How is the trend towards bicycle use by both types of customers (registered & casual customers) during the loan period from January 1, 2011 - December 31, 2012 based on working days and weekend/holiday?





The results of the two line charts above give us information that in the 2 year period (1 January 2011 to 31 December 2012) the number of users has always experienced a significant increase, both casual and registered users in the GowesKuyy Company's automated system

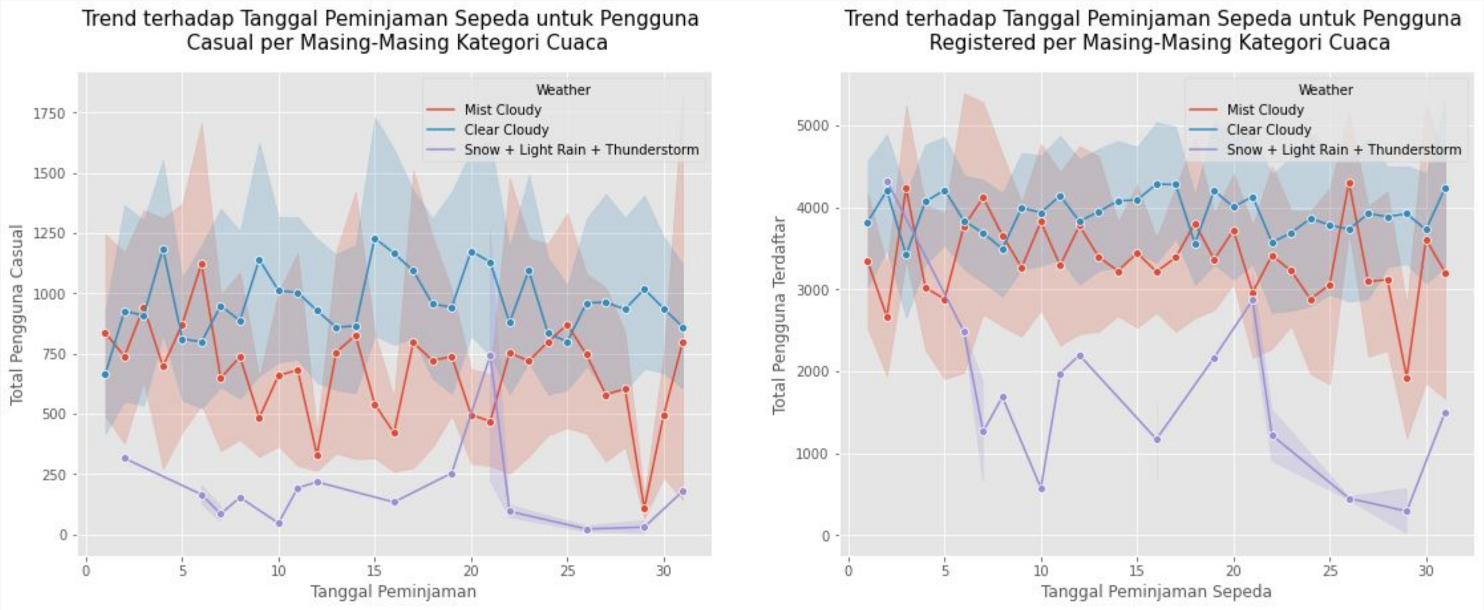
Fourth Question: What is the average number of bicycle users each day?



The bar chart on the side shows that Friday is the day that has the highest average number of users compared to other days with a total average value of 4,690 people. Meanwhile, Sunday is the day that has the lowest average user value among the six other days because it is only 4,229 people

This insight is very surprising because Sunday is a relaxing day for some people so it is very suitable for cycling activities. But in reality, Sundays produce the lowest average bike usage in 2 years

Fifth Question: What are the Trends in Bicycle Rental Use from Registered and Casual Users based on Rental per Day in each Weather Category?



The trend of using bicycles when the weather is snowy is very low and the number of users is very small compared to the number of users when the weather is clear and mist cloudy. This is understandable because the majority of people in the world will not want to cycle when the weather outside is experiencing a thunderstorm or snow

06

Conclusion

Conclusion based on Five Analysis Questions

01

Autumn in 2011 and 2012 were the seasons that had the highest total number of bicycle users among the three other seasons, namely spring, summer and winter

02

The majority of bicycle use occurs on weekdays rather than weekends/holidays

03

Over a period of 2 years by providing bicycle rental services, GowesKuyy Company has succeeded in generating an increase in both casual and registered users.

04

Friday had the highest average number of users among the six other days, namely 4,690 people. Meanwhile, Sunday is the day that has the lowest average user value among the other six days because it is only 4,229 people

05

Snowy Weather has the least number of users compared to Clear & Mist Seasons based on the date of use of the rental bike

Source of Bike Sharing Dataset

https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset

Find Me At

ekapramudianzah554@gmail.com

https://github.com/pramudianzaheka92

https://www.linkedin.com/in/eka-pramudianzah/