Universidad del Valle de Guatemala
Minería de Datos
S20

Hoja de Trabajo #2

EDA y Regresiones

Javier Chavez 21016
Andres Quezada 21085

Parte 1 – Análisis exploratorio de datos

1.1. Haga una exploración rápida de sus datos, haga un resumen del conjunto de datos.

Descripción de las variables:

attendance: Número de espectadores que asistieron al juego.
away_team: Nombre del equipo visitante.
away_team_errors: Número de errores cometidos por el equipo visitante.
away_team_hits: Número de hits conseguidos por el equipo visitante.
away_team_runs: Número de carreras anotadas por el equipo visitante.
boxscore_url: URL al boxscore del juego en Baseball-Reference.
date: Fecha en la que se jugó el partido.
field_type: Valores nulos
game_duration: Duración del juego.
game_type: Tipo de juego, por ejemplo, si es un juego nocturno y si se juega en césped.
home_team: Nombre del equipo local.
home_team_errors: Número de errores cometidos por el equipo local.
home_team_hits: Número de hits conseguidos por el equipo local.
home_team_runs: Número de carreras anotadas por el equipo local.
other_info_string: Información ajena
start_time: Hora de inicio del juego.
venue: Estadio donde se jugó el partido.

Describe:

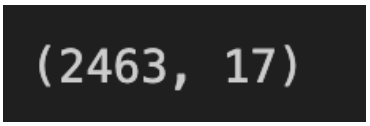
	away_team_errors	away_team_hits	away_team_runs	field_type	home_team_errors	home_team_hits	home_team_runs
count	2463.000000	2463.000000	2463.000000	0.0	2463.000000	2463.000000	2463.000000
mean	0.580593	8.764515	4.413723	NaN	0.585871	8.611855	4.519691
std	0.793391	3.511581	3.104556	NaN	0.805542	3.436965	3.111572
min	0.000000	1.000000	0.000000	NaN	0.000000	0.000000	0.000000
25%	0.000000	6.000000	2.000000	NaN	0.000000	6.000000	2.000000
50%	0.000000	8.000000	4.000000	NaN	0.000000	8.000000	4.000000
75%	1.000000	11.000000	6.000000	NaN	1.000000	11.000000	6.000000
max	5.000000	22.000000	21.000000	NaN	5.000000	22.000000	17.000000

Info:

```
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   attendance         2463 non-null   object
 1   away_team          2463 non-null   object
 2   away_team_errors   2463 non-null   int64
 3   away_team_hits     2463 non-null   int64
 4   away_team_runs     2463 non-null   int64
 5   boxscore_url       2463 non-null   object
 6   date               2463 non-null   object
 7   field_type         0 non-null      float64
 8   game_duration      2463 non-null   object
 9   game_type          2460 non-null   object
 10  home_team          2463 non-null   object
 11  home_team_errors   2463 non-null   int64
 12  home_team_hits     2463 non-null   int64
 13  home_team_runs     2463 non-null   int64
 14  other_info_string  2463 non-null   object
 15  start_time         2463 non-null   object
 16  venue              2463 non-null   object
dtypes: float64(1), int64(6), object(10)
```

Shape:

2463 filas y 17 columnas



Variables inútiles para el análisis de datos:

- boxscore_url: El URL no aporta nada a la investigación
- other_info_string: son comentarios extras diferentes en cada utilización.

Variables que necesitan transformación:

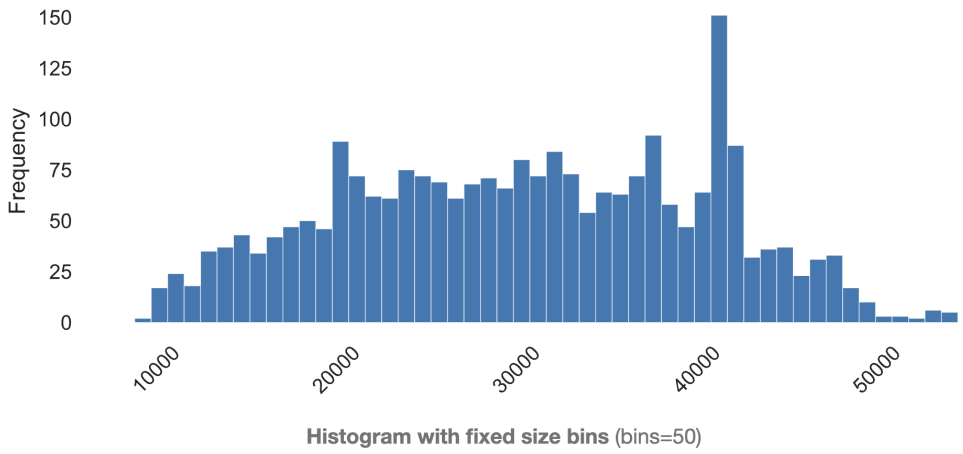
- **attendance:** tiene carácteres extras
- **date:** estandarizar la fecha
- **game_duration:** estandarizar los datos
- **start_time:** estandarizar los datos
- **field_type**: extraer el field_type de la columna game_type.

1.2. Determine el tipo de cada una de las variables.

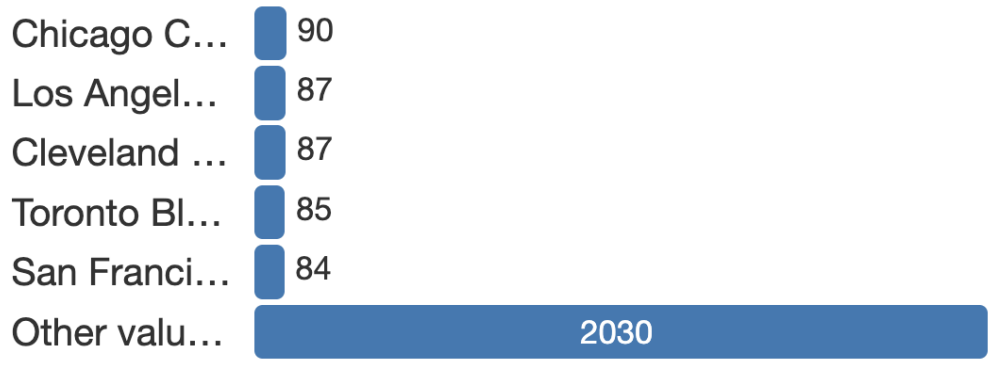
Variable	Tipo de Variable
attendance	Cuantitativa Discreta
away_team	Cualitativa Nominal
away_team_errors	Cuantitativa Discreta
away_team_hits	Cuantitativa Discreta
away_team_runs	Cuantitativa Discreta
date	Cualitativa Ordinal
game_duration	Cuantitativa Continua
game_type	Cualitativa Nominal
home_team	Cualitativa Nominal
home_team_errors	Cuantitativa Discreta
home_team_hits	Cuantitativa Discreta
home_team_runs	Cuantitativa Discreta
start_time	Cualitativa Ordinal
venue	Cualitativa Nominal

1.3. Incluya los gráficos exploratorios siendo consecuentes con el tipo de variable que están representando.

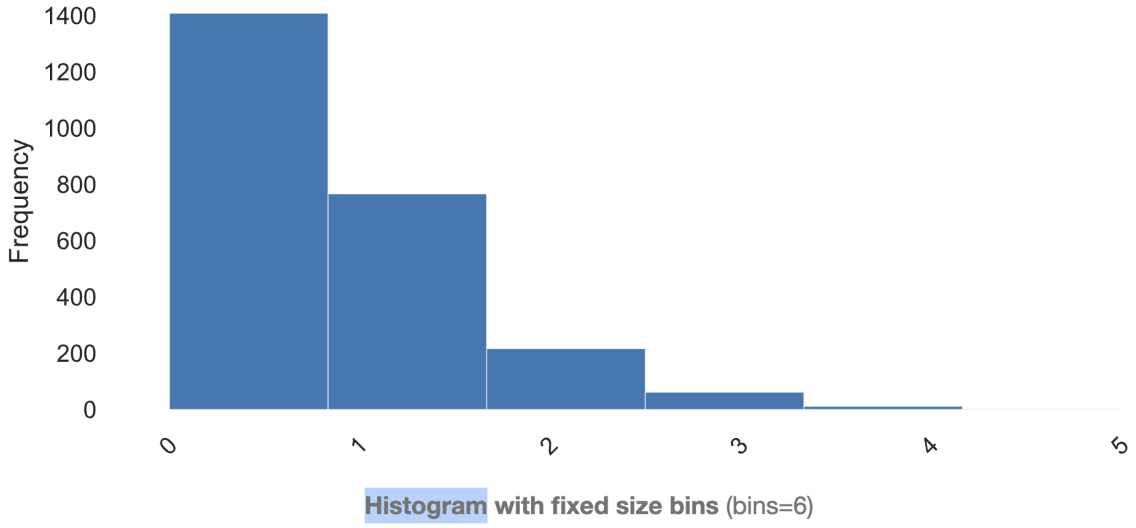
Attendance



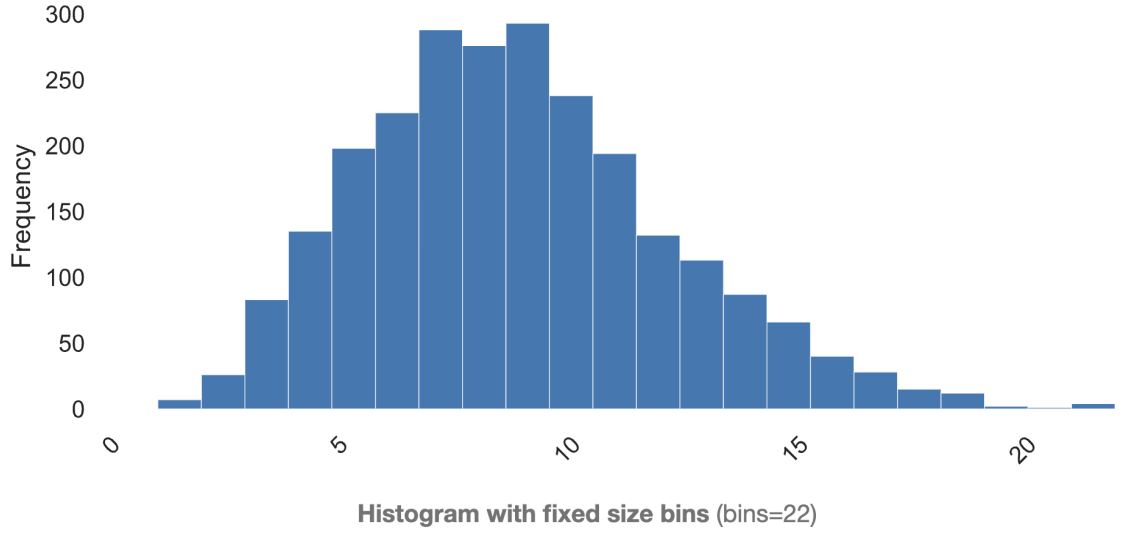
Away_team



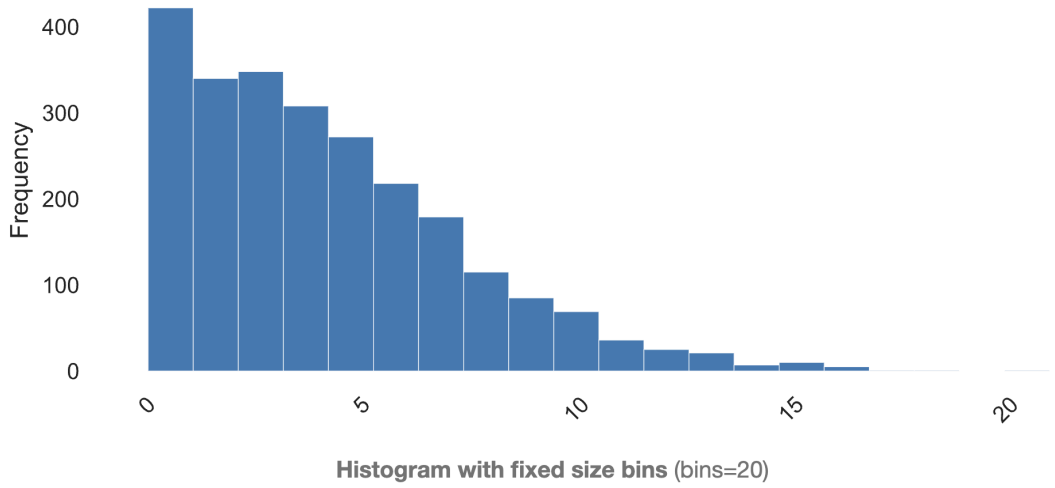
away



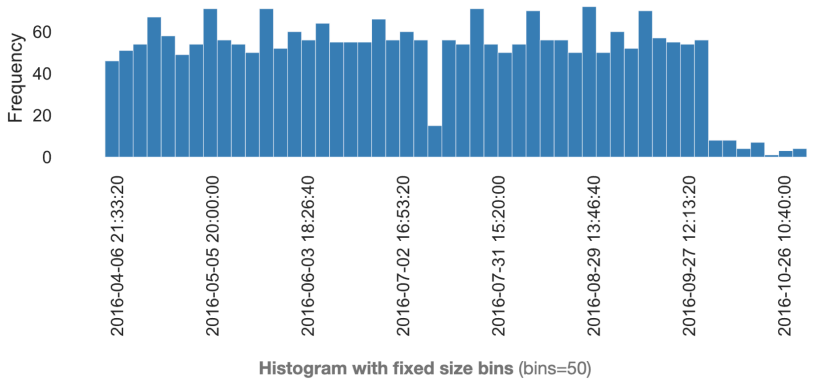
away_team_hits



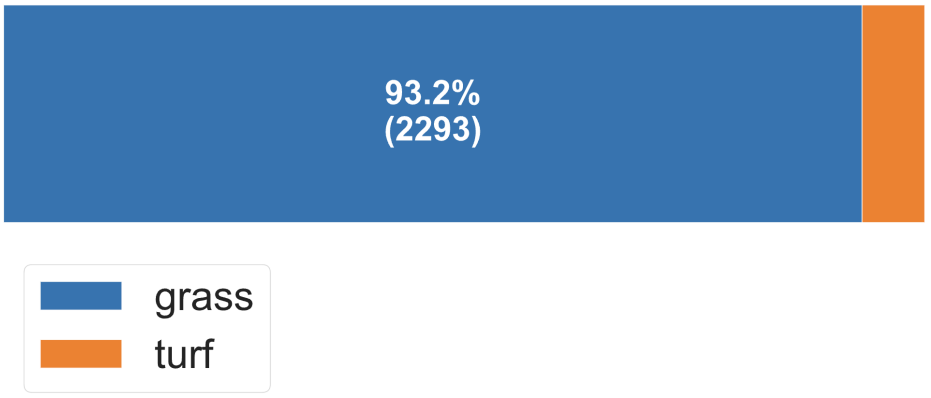
away_team_runs



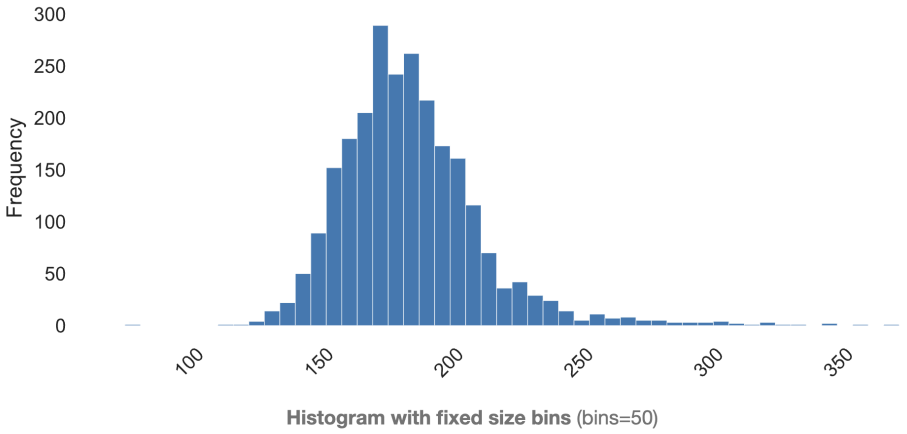
date



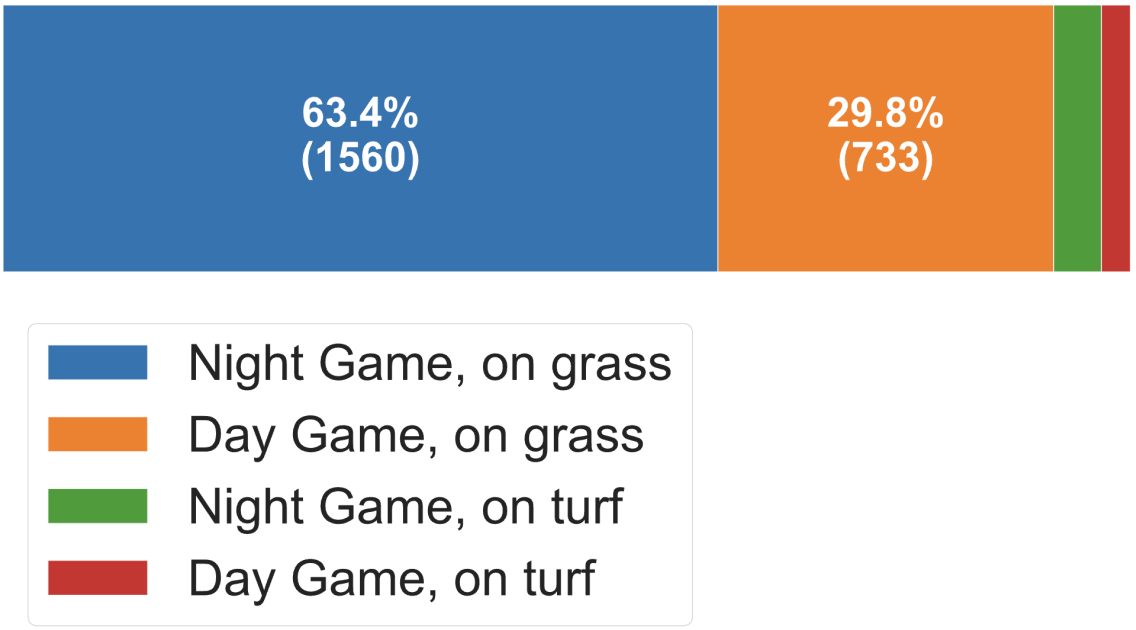
field_type



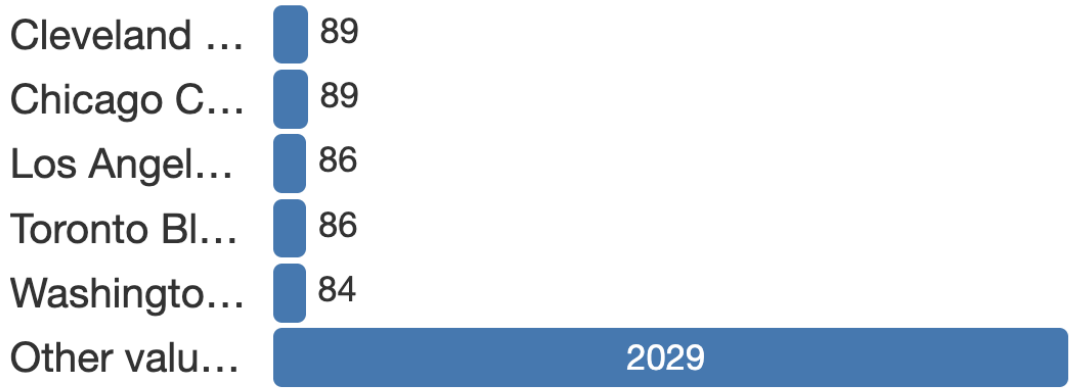
game_duration



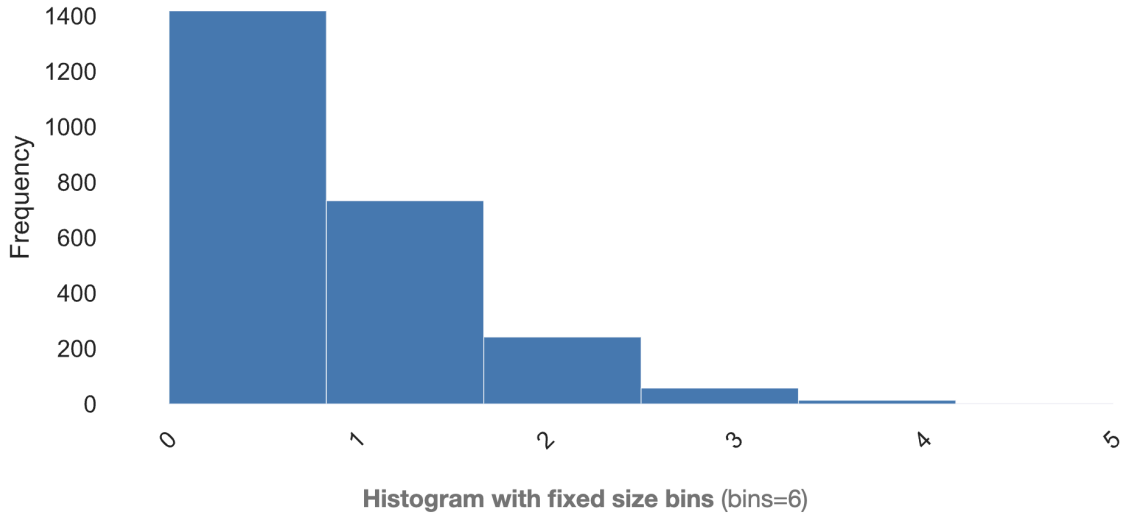
game_type



home_team

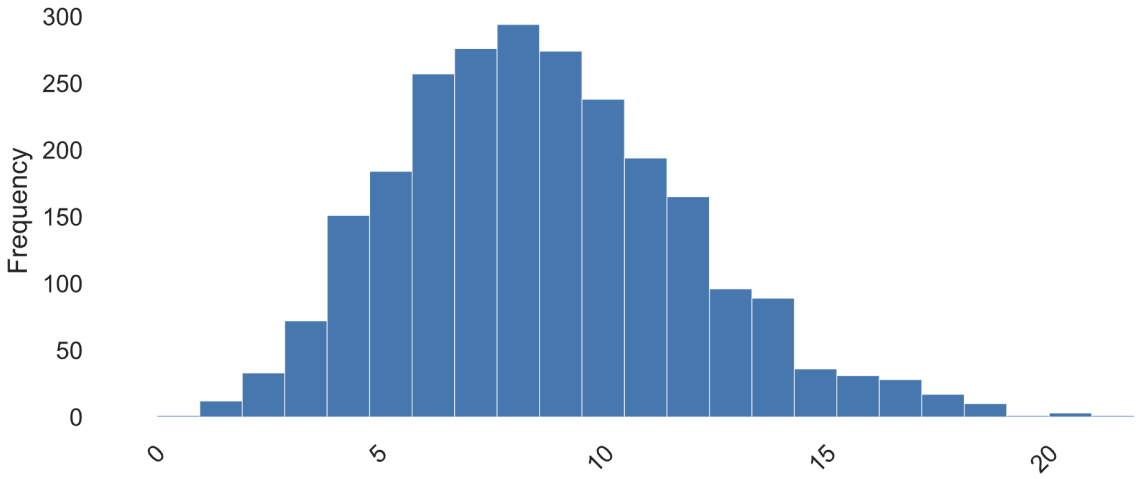


home_team_errors



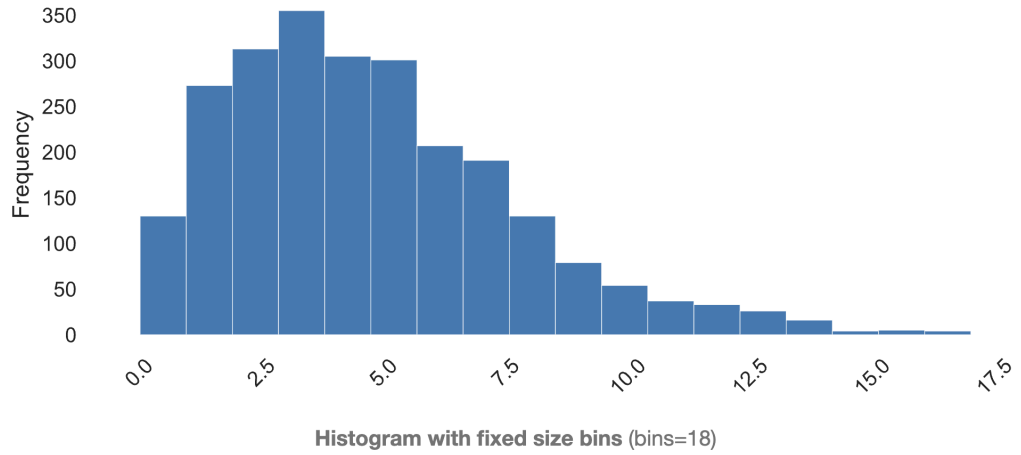
Histogram with fixed size bins (bins=6)

home_team_hits

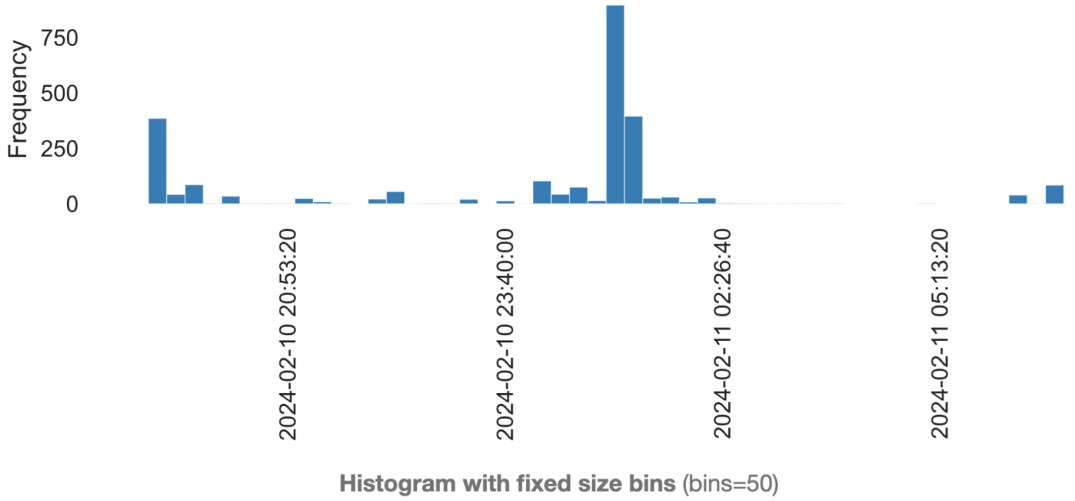


Histogram with fixed size bins (bins=23)

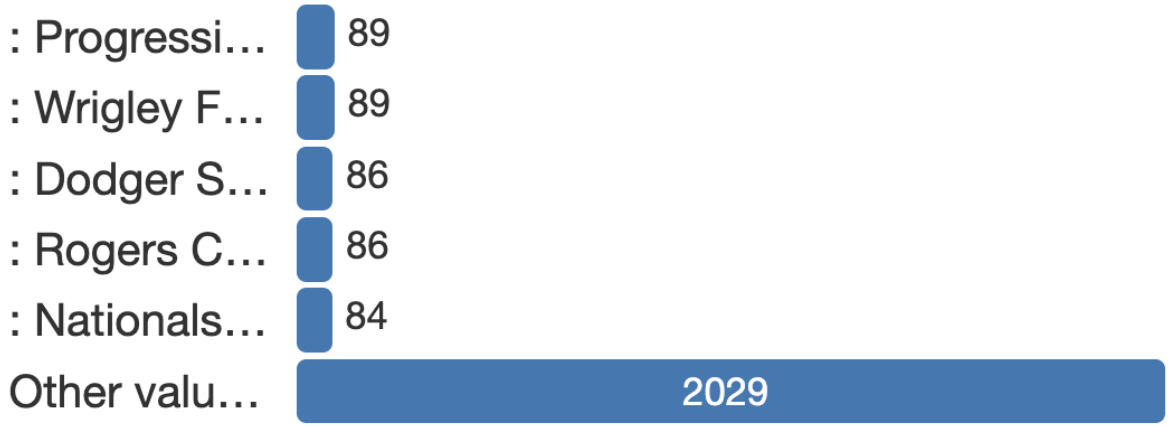
home_team_runs



start_time

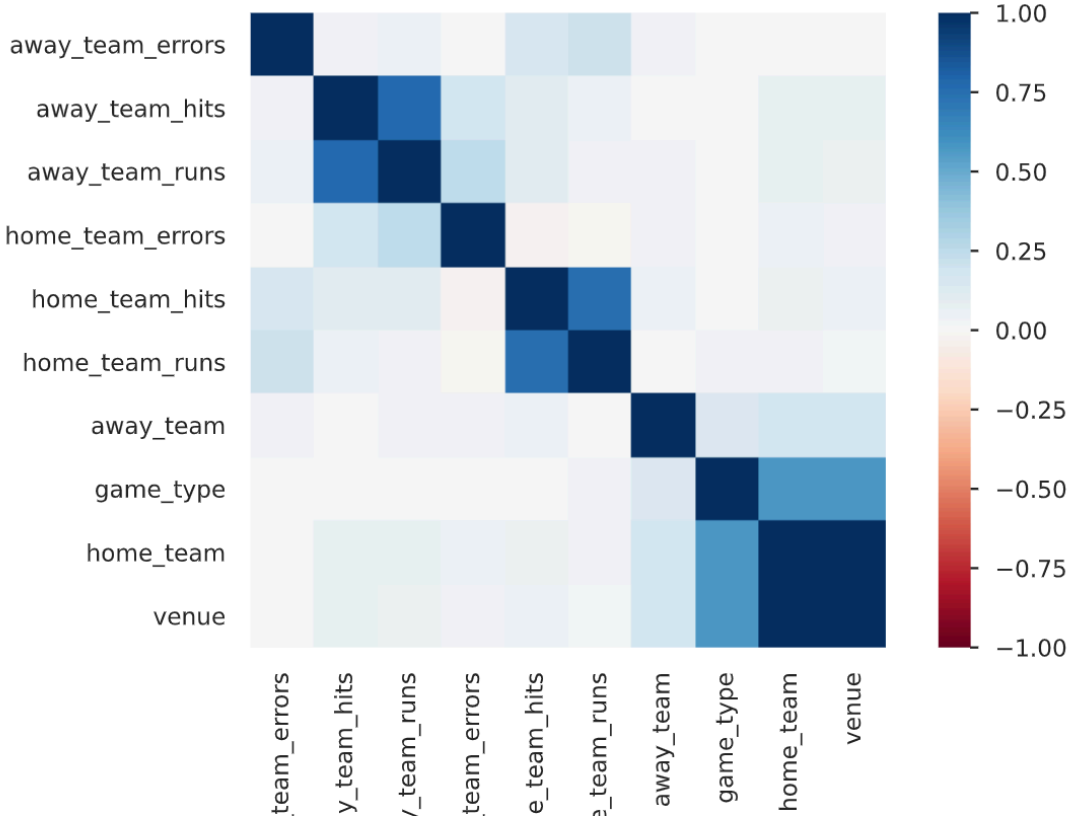


venue



1.4. Aísle las variables numéricas de las categóricas, haga un análisis de correlación entre las mismas.

Matriz de correlación entre las variables numéricas

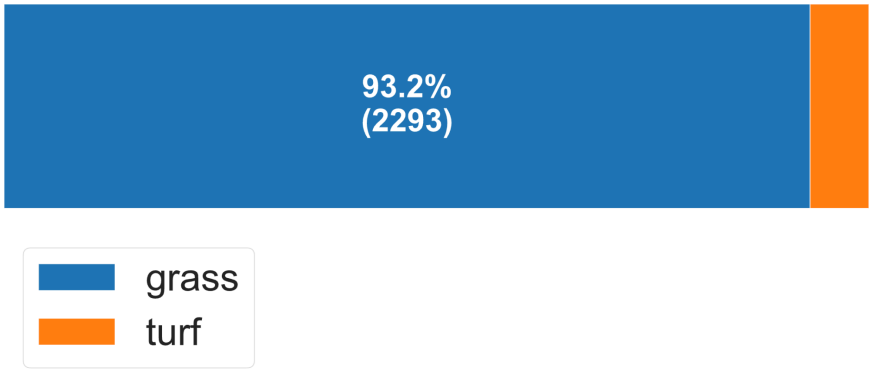


1.5. Utilice las variables categóricas, haga tablas de frecuencia, proporción, gráficas de barras o cualquier otra técnica que le permita explorar los datos.

Categoría: away_team

Value	Count	Frequency (%)
Chicago Cubs	90	3.7%
Los Angeles Dodgers	87	3.5%
Cleveland Indians	87	3.5%
Toronto Blue Jays	85	3.5%
San Francisco Giants	84	3.4%
Boston Red Sox	83	3.4%
Washington Nationals	83	3.4%
Baltimore Orioles	82	3.3%
Texas Rangers	82	3.3%
St. Louis Cardinals	81	3.3%
Other values (20)	1619	65.7%

Categoría: field_type



Categoría: home_team

Value	Count	Frequency (%)
Cleveland Indians	89	3.6%
Chicago Cubs	89	3.6%
Los Angeles Dodgers	86	3.5%
Toronto Blue Jays	86	3.5%
Washington Nationals	84	3.4%
Texas Rangers	83	3.4%
San Francisco Giants	83	3.4%
New York Mets	82	3.3%
Boston Red Sox	82	3.3%
Tampa Bay Rays	81	3.3%
Other values (20)	1618	65.7%

1.6. Realice la limpieza de variables utilizando las técnicas vistas en clase, u otras que piense pueden ser de utilidad

```
# Limpiando la columna 'attendance'
df['attendance'] = df['attendance'].str.replace(',', '')
df['attendance'] = df['attendance'].str.replace(']', '')
df['attendance'] = df['attendance'].str.replace("'", '')
df['attendance'] = df['attendance'].str.replace(' ', '')
df['attendance'] = pd.to_numeric(df['attendance'], errors='coerce')


# Mostrar las primeras filas del DataFrame limpio
print(df.head())
```

```
# Limpiar start time
df['start_time'] = df['start_time'].str.extract(r'(\d+:\d+)')

# Mostar las primeras filas del DataFrame limpio
print(df.head())
```

```
# limpiar game_duration
df['game_duration'] = df['game_duration'].str.replace(":", "")
df['game_duration'] = pd.to_numeric(df['game_duration'], errors='coerce')
df['game_duration'] = (df['game_duration'] // 100) * 60 + (df['game_duration'] % 100)

# mostrar las primeras filas del DataFrame limpio
print(df.head())
```

✓ 0.0s

```
# Limpar game_type y llenar field_type
df['field_type'] = df['game_type'].str.extract(r'on\s+(\w+)')

# Mostar las primeras filas del DataFrame limpio
print(df.head())
```

Parte 2 – Pruebe todos los modelos de Regresión vistos en clase y encuentre el mejor modelo de ellos para predecir el número de asistentes a un partido. Para cada modelo:

2.1. Siga los procedimientos vistos en clase para realizar una regresión con los datos dados

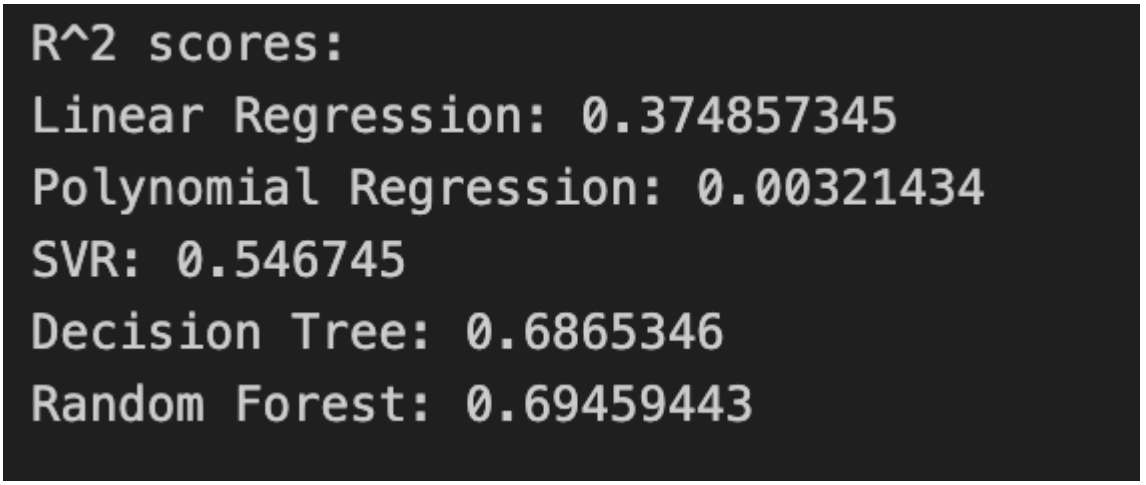
Archivo adjunto.py

```python
# Seleccionar las nuevas variables independientes
X = df[[ 'start_time', 'date', 'venue', 'away_team', 'home_team']]
y = df['attendance']

# Convertir variables categóricas en variables dummy
X = pd.get_dummies(X)

# Dividir los datos en conjuntos de entrenamiento y prueba
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Entrenar modelos de regresión
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score

# Inicializar los modelos de regresión
regressors = {
    'Linear Regression': LinearRegression(),
    'Polynomial Regression': PolynomialFeatures(degree=2),
    'SVR': SVR(),
    'Decision Tree': DecisionTreeRegressor(),
    'Random Forest': RandomForestRegressor()
}

# Entrenar los modelos y calcular R^2 score
results = {}
for name, regressor in regressors.items():
    if name == 'Polynomial Regression':
        X_poly_train = regressor.fit_transform(X_train)
        X_poly_test = regressor.transform(X_test)
        regressor = LinearRegression()
        regressor.fit(X_poly_train, y_train)
        y_pred = regressor.predict(X_poly_test)
    else:
        regressor.fit(X_train, y_train)
        y_pred = regressor.predict(X_test)
    results[name] = r2_score(y_test, y_pred)
```

2.2. ¿Cuál es el rendimiento de su modelo? Calcule el parámetro R2 para dar respaldo a su respuesta

Al evaluar el rendimiento de los modelos creados, podemos observar que los de mejor rendimiento son los de Decision Tree y Random Forest. Teniendo un valor aproximado de 0,7 lo cual se puede calificar como bueno. Mientras que los otros modelos no se ajustan correctamente.

Esto es correcto ya que el modelo de Random Forest se destaca como una opción óptima para el conjunto de datos en cuestión debido a su capacidad para manejar eficazmente características categóricas sin necesidad de codificación adicional. Además, su naturaleza de conjunto de árboles de decisión proporciona una robustez inherente contra el sobreajuste, lo que facilita la generalización a datos nuevos. La flexibilidad del algoritmo permite capturar relaciones complejas y no lineales entre las variables de entrada y la variable objetivo, mientras que su escalabilidad garantiza un rendimiento eficiente incluso en conjuntos de datos grandes



2.3. Si es uno de los modelos lineales, obtenga las constantes del modelo y exprese la ecuación que representan

Attendance = β0 + β1 * start_time + β2 * date + β3 * venue + ...

Donde:

β0 es el intercepto.
β1, β2, β3 son los coeficientes asociados con las variables independientes start_time, date, venue respectivamente.

Con:
Intercept: 1.5987752559175776e+17

Y coeficientes:

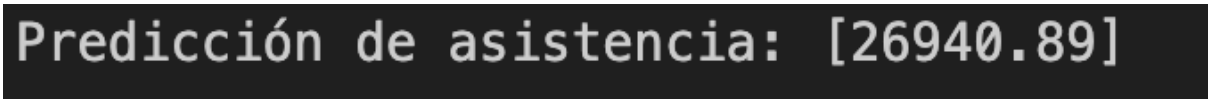
```
Constantes del modelo de regresión lineal:
Coefficients: [-2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
  2.60039955e+17 -2.05806923e+15  1.36122828e+17 -2.05806923e+15
 -2.05806923e+15  2.22474249e+16 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15  3.65640210e+16 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15 -2.05806923e+15
```

2.4. Está interesado en predecir cuál será la asistencia a un partido en el que se enfrenten X y Y equipos (Ud decide cuáles), así como el día de la semana, la hora y el estadio (también los decide Ud) y otras variables que exija su modelo. Para estos valores, ¿cuál es la predicción de la asistencia?

Para un partido con estas características:

```
new_data = {
    'start_time': '7:00 PM',
    'date': '2024-02-15',
    'venue': 'Great American Ball Park',
    'away_team': 'Kansas City Royals',
    'home_team': 'Cincinnati Reds'
}
```

Y utilizando el mejor modelo encontrado (Random Forest, la predicción de asistencia es de 26,940 personas.



Conclusiones:

- El mejor modelo de predicción para el conjunto de datos presente es el Random Forest, debido a su capacidad para manejar eficientemente características categóricas y su robustez contra el sobreajuste.

- La predicción de la asistencia para el partido entre Kansas City y Cincinnati Reds es de 26,940 personas, proporcionando una estimación útil para la planificación logística del evento.

- La limpieza de datos es de suma importancia para garantizar que el modelo se entrene con datos de alta calidad, lo que contribuye a la precisión y la confiabilidad de las predicciones.

- Es fundamental reconocer que los diferentes modelos de predicción tienen diferentes ventajas y desventajas. Por lo tanto, es necesario revisar el coeficiente de determinación R^2 para determinar qué modelo se ajusta mejor a la situación específica antes de tomar decisiones basadas en las predicciones.

Referencias

Equipo de desarrollo de Matplotlib. (2023). Matplotlib: A Comprehensive Library for Creating Static, Animated, and Interactive Visualizations in Python [Software]. Disponible en https://matplotlib.org

McKinney, W., & otros colaboradores de Pandas. (2023). pandas: Powerful data structures for data analysis, time series, and statistics [Software]. Disponible en https://pandas.pydata.org

Waskom, M. (2023). Seaborn: Statistical Data Visualization [Software]. Disponible en https://seaborn.pydata.org