In [1]:
```
import pandas as pd
```

In [2]:
```
draft_df = pd.read_csv(r"C:\Users\Omie\Desktop\DSC 530 Project\nfl_draft_prospects.csv")
```

In [3]:
```
print("Draft Data:")
print(draft_df.head())
```

```
Draft Data:
   draft_year  player_id     player_name       position pos_abbr  \
0        1967      23590     Bubba Smith  Defensive End       DE
1        1967      23591   Clinton Jones   Running Back       RB
2        1967      23592  Steve Spurrier    Quarterback       QB
3        1967      23593      Bob Griese    Quarterback       QB
4        1967      23594  George Webster     Linebacker       LB

           school   school_name school_abbr  \
0  Michigan State      Spartans         MSU
1  Michigan State      Spartans         MSU
2         Florida        Gators         FLA
3          Purdue  Boilermakers         PUR
4  Michigan State      Spartans         MSU

                                            link  pick  ...  \
0  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   1.0  ...
1  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   2.0  ...
2  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   3.0  ...
3  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   4.0  ...
4  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   5.0  ...

                 team  team_abbr  \
0     Baltimore Colts         IND
1   Minnesota Vikings         MIN
2  San Francisco 49ers          SF
3      Miami Dolphins         MIA
4      Houston Oilers         TEN

                                    team_logo_espn guid weight height  \
0  https://a.espncdn.com/i/teamlogos/nfl/500/scor... (https://a.espncdn.com/i/teamlogos/nfl/500/scor...)  NaN    NaN    NaN
1  https://a.espncdn.com/i/teamlogos/nfl/500/scor... (https://a.espncdn.com/i/teamlogos/nfl/500/scor...)  NaN    NaN    NaN
2  https://a.espncdn.com/i/teamlogos/nfl/500/scor... (https://a.espncdn.com/i/teamlogos/nfl/500/scor...)  NaN    NaN    NaN
3  https://a.espncdn.com/i/teamlogos/nfl/500/scor... (https://a.espncdn.com/i/teamlogos/nfl/500/scor...)  NaN    NaN    NaN
4  https://a.espncdn.com/i/teamlogos/nfl/500/scor... (https://a.espncdn.com/i/teamlogos/nfl/500/scor...)  NaN    NaN    NaN

  pos_rk ovr_rk  grade  player_image
0    NaN    NaN    NaN           NaN
1    NaN    NaN    NaN           NaN
2    NaN    NaN    NaN           NaN
3    NaN    NaN    NaN           NaN
4    NaN    NaN    NaN           NaN

[5 rows x 24 columns]
```

In [4]:
```
performance_df = pd.read_csv(r"C:\Users\Omie\Desktop\DSC 530 Project\yearly_player_data.csv")
```

In [5]:
```
print("Performance Data:")
print(performance_df.head())
```

```
Performance Data:
  team   player_id player_name position  season  depth  pass_attempts  \
0  TEN  00-0035676  A.J. Brown       WR    2019    2.0            0.0
1  TEN  00-0035676  A.J. Brown       WR    2020    1.0            0.0
2  TEN  00-0035676  A.J. Brown       WR    2021    1.0            2.0
3  PHI  00-0035676  A.J. Brown       WR    2022    1.0            0.0
4  PHI  00-0035676  A.J. Brown       WR    2023    1.0            0.0

   complete_pass  incomplete_pass  passing_yards  ...  vacated_receptions  \
0            0.0              0.0            0.0  ...               147.0
1            0.0              0.0            0.0  ...                62.0
2            0.0              2.0            0.0  ...                74.0
3            0.0              0.0            0.0  ...               135.0
4            0.0              0.0            0.0  ...                47.0

   vacated_receiving_yards  vacated_receiving_air_yards  \
0                   1632.0                       1886.0
1                    730.0                       1015.0
2                    741.0                        804.0
3                   1769.0                       2911.0
4                    471.0                        753.0

   vacated_yards_after_catch  vacated_reception_td  vacated_rush_attempts  \
0                      646.0                   6.0                  185.0
1                      284.0                   4.0                    8.0
2                      331.0                   7.0                   88.0
3                      463.0                  10.0                   83.0
4                      217.0                   6.0                   96.0

   vacated_rushing_yards  vacated_run_td  vacated_touches  vacated_total_yards
0                  656.0             6.0            383.0               2420.0
1                   19.0             0.0            176.0               1338.0
2                  365.0             0.0            345.0               2147.0
3                  397.0             6.0            724.0               4486.0
4                  438.0             3.0            145.0                911.0

[5 rows x 195 columns]
```

In [6]:
```
# Display column names
print("Draft Dataset Columns:", draft_df.columns)
print("Performance Dataset Columns:", performance_df.columns)
```

```
Draft Dataset Columns: Index(['draft_year', 'player_id', 'player_name', 'position', 'pos_abbr',
       'school', 'school_name', 'school_abbr', 'link', 'pick', 'overall',
       'round', 'traded', 'trade_note', 'team', 'team_abbr', 'team_logo_espn',
       'guid', 'weight', 'height', 'pos_rk', 'ovr_rk', 'grade',
       'player_image'],
      dtype='object')
Performance Dataset Columns: Index(['team', 'player_id', 'player_name', 'position', 'season', 'depth',
       'pass_attempts', 'complete_pass', 'incomplete_pass', 'passing_yards',
       ...
       'vacated_receptions', 'vacated_receiving_yards',
       'vacated_receiving_air_yards', 'vacated_yards_after_catch',
       'vacated_reception_td', 'vacated_rush_attempts',
       'vacated_rushing_yards', 'vacated_run_td', 'vacated_touches',
       'vacated_total_yards'],
      dtype='object', length=195)
```

In [7]:
```
# Convert names to lowercase and strip spaces
draft_df["player_name"] = draft_df["player_name"].str.lower().str.strip()
performance_df["player_name"] = performance_df["player_name"].str.lower().str.strip()
```

In [8]:
```
# Merge datasets on player_name
merged_df = pd.merge(draft_df, performance_df, on="player_name", how="inner")

# Display merged dataset
print(merged_df.head())
```

```
   draft_year_x  player_id_x     player_name     position_x pos_abbr  \
0          1967        23681       tim jones    Quarterback       QB
1          1967        23681       tim jones    Quarterback       QB
2          2021       105442       tim jones  Wide Receiver       WR
3          2021       105442       tim jones  Wide Receiver       WR
4          1967        12413  william powell     Linebacker       LB

                school    school_name school_abbr  \
0          Weber State       Wildcats         WEB
1          Weber State       Wildcats         WEB
2  Southern Mississippi  Golden Eagles         USM
3  Southern Mississippi  Golden Eagles         USM
4             Missouri         Tigers         MIZ

                                          link  pick  ...  \
0  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   8.0  ...
1  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   8.0  ...
2  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   NaN  ...
3  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)   NaN  ...
4  http://insider.espn.com/nfl/draft/player/_/id/... (http://insider.espn.com/nfl/draft/player/_/id/...)  25.0  ...

   vacated_receptions  vacated_receiving_yards vacated_receiving_air_yards  \
0               187.0                   1988.0                      2705.0
1               223.0                   2164.0                      2396.0
2               187.0                   1988.0                      2705.0
3               223.0                   2164.0                      2396.0
4                 NaN                      NaN                         NaN

  vacated_yards_after_catch vacated_reception_td vacated_rush_attempts  \
0                     688.0                12.0                  44.0
1                    1189.0                 7.0                 310.0
2                     688.0                12.0                  44.0
3                    1189.0                 7.0                 310.0
4                       NaN                 NaN                   NaN

  vacated_rushing_yards vacated_run_td  vacated_touches  vacated_total_yards
0                 213.0           2.0            895.0               5908.0
1                1302.0          10.0            533.0               3466.0
2                 213.0           2.0            895.0               5908.0
3                1302.0          10.0            533.0               3466.0
4                   NaN           NaN              NaN                  NaN

[5 rows x 218 columns]
```

In [9]:
```
print(merged_df.isnull().sum())
```

```
draft_year_x               0
player_id_x                0
player_name                0
position_x                 0
pos_abbr                   0
                        ...
vacated_rush_attempts    345
vacated_rushing_yards    345
vacated_run_td           345
vacated_touches          345
vacated_total_yards      345
Length: 218, dtype: int64
```

In [10]:
```
# Fill missing performance metrics with 0
merged_df.fillna(0, inplace=True)
```

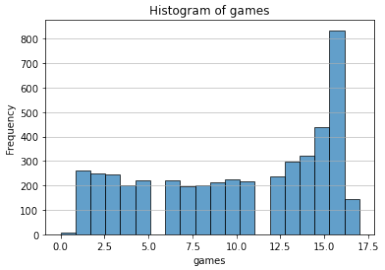
In [11]:
```
merged_df.to_csv("merged_nfl_draft_data.csv", index=False)
```

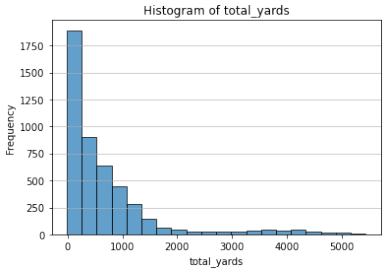
In [12]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import ttest_ind
import statsmodels.api as sm
```

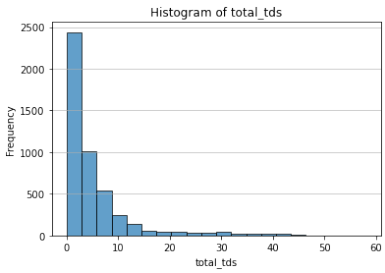
In [13]:
```
df = pd.read_csv(r"C:\Users\Omie\Desktop\DSC 530 Project\merged_nfl_draft_data.csv", low_memory=False)
```

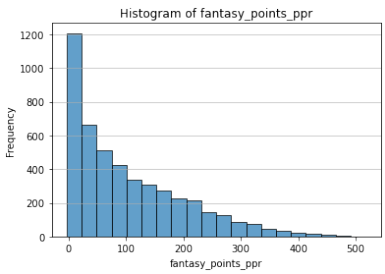
In [14]:
```
# Select key variables
variables = ['games', 'total_yards', 'total_tds', 'fantasy_points_ppr', 'round']
```

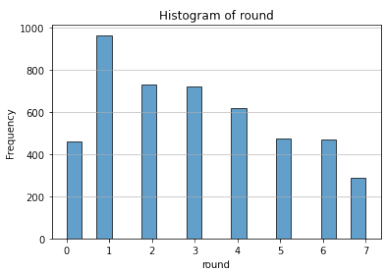
In [15]:
```
# Histograms
variables = ['games', 'total_yards', 'total_tds', 'fantasy_points_ppr', 'round']

for var in variables:
    plt.figure(figsize=(6, 4))
    plt.hist(df[var], bins=20, edgecolor='black', alpha=0.7)
    plt.title(f'Histogram of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.grid(axis='y', alpha=0.75)
    plt.show()
```



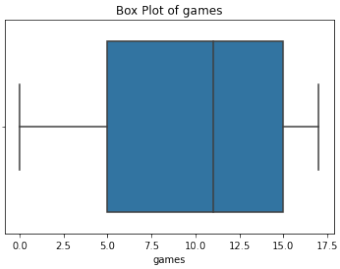


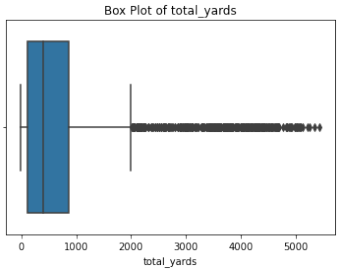


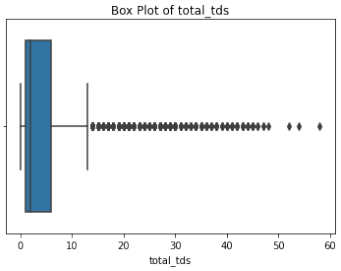


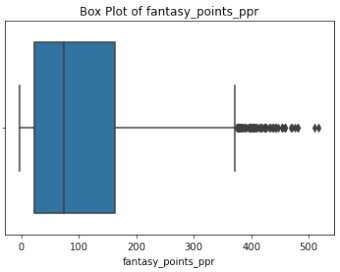


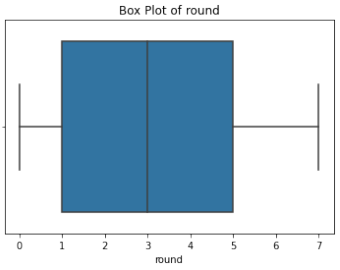
In [16]:
```
# Boxplots for outliers
for var in variables:
    plt.figure(figsize=(6, 4))
    sns.boxplot(x=df[var])
    plt.title(f'Box Plot of {var}')
    plt.show()
```











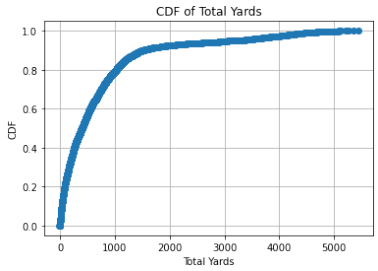
In [17]:
```
# Summary statistics
summary_stats = df[variables].describe().T
summary_stats['mode'] = df[variables].mode().iloc[0]
summary_stats['spread'] = summary_stats['max'] - summary_stats['min']

print(summary_stats)

summary_stats.to_csv("summary_statistics.csv")
```

```
                    count        mean         std    min     25%    50%  \
games              4731.0   10.107166    5.165192   0.00    5.00   11.0
total_yards        4731.0  705.918622  949.508775 -14.00  118.50  399.0
total_tds          4731.0    4.945043    7.499538   0.00    1.00    2.0
fantasy_points_ppr 4731.0  103.849622   97.980116  -3.32   22.05   73.9
round              4731.0    3.015853    2.059671   0.00    1.00    3.0

                     75%      max  mode  spread
games               15.0    17.00  16.0    17.0
total_yards        872.0  5440.00   0.0  5454.0
total_tds            6.0    58.00   0.0    58.0
fantasy_points_ppr 162.5   517.38   0.0   520.7
round                5.0     7.00   1.0     7.0
```

In [18]:
```
# Probability Mass Function
def compute_pmf(data):
    counts = data.value_counts(normalize=True)
    return counts.sort_index()

# Compute PMF for first-round vs. later rounds
first_round_pmf = compute_pmf(df[df['round'] == 1]['games'])
later_round_pmf = compute_pmf(df[df['round'] > 1]['games'])

# Display PMF results
print("First Round PMF:\n", first_round_pmf.head())
print("Later Round PMF:\n", later_round_pmf.head())
```

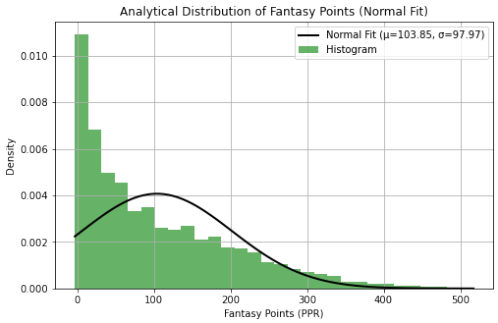
```
First Round PMF:
 1    0.021739
2    0.025880
3    0.034161
4    0.021739
5    0.027950
Name: games, dtype: float64
Later Round PMF:
 0    0.001816
1    0.055387
2    0.052361
3    0.056598
4    0.046005
Name: games, dtype: float64
```

In [19]:
```
# Cumulative Distribution Function
def compute_cdf(data):
    sorted_data = np.sort(data)
    cdf = np.arange(1, len(sorted_data) + 1) / len(sorted_data)
    return sorted_data, cdf

x, y = compute_cdf(df['total_yards'])

plt.figure(figsize=(6, 4))
plt.plot(x, y, marker='o', linestyle='none')
plt.xlabel('Total Yards')
plt.ylabel('CDF')
plt.title('CDF of Total Yards')
plt.grid()
plt.show()
```



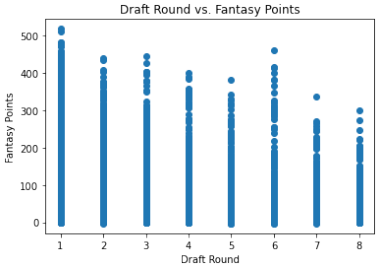
In [20]:
```
import scipy.stats as stats
```

In [21]:
```
data = df['fantasy_points_ppr'].dropna()

# Fit a normal distribution to the data
mu, std = stats.norm.fit(data)

# Generate values for plotting the fitted distribution
xmin, xmax = data.min(), data.max()
x = np.linspace(xmin, xmax, 100)
pdf = stats.norm.pdf(x, mu, std)

# Plot histogram and fitted normal distribution
plt.figure(figsize=(8, 5))
plt.hist(data, bins=30, density=True, alpha=0.6, color='g', label="Histogram")
plt.plot(x, pdf, 'k', linewidth=2, label=f"Normal Fit (μ={mu:.2f}, σ={std:.2f})")

plt.title("Analytical Distribution of Fantasy Points (Normal Fit)")
plt.xlabel("Fantasy Points (PPR)")
plt.ylabel("Density")
plt.legend()
plt.grid()
plt.show()

# Perform normality test (Shapiro-Wilk Test)
shapiro_test_stat, shapiro_p_value = stats.shapiro(data.sample(500, random_state=42)) if len(data) > 500 else stats.shapiro(data)

# Store results in a DataFrame
distribution_results = {
    "Mean (μ)": mu,
    "Standard Deviation (σ)": std,
    "Shapiro-Wilk Test Statistic": shapiro_test_stat,
    "Shapiro-Wilk P-Value": shapiro_p_value
}

# Convert to DataFrame for display
distribution_df = pd.DataFrame([distribution_results])

print(distribution_df.to_string(index=False))
```

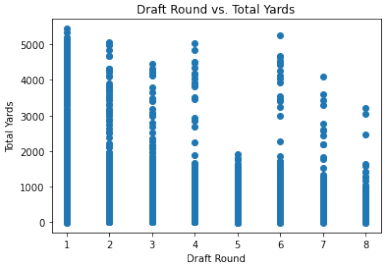



```
  Mean (μ)  Standard Deviation (σ)  Shapiro-Wilk Test Statistic  Shapiro-Wilk P-Value
103.849622                97.96976                     0.871248          6.602537e-20
```

In [22]:
```
# Scatter Plot: Draft Round vs. Fantasy Points
plt.figure(figsize=(6, 4))
plt.scatter(df['draft_round'], df['fantasy_points_ppr'])
plt.xlabel('Draft Round')
plt.ylabel('Fantasy Points')
plt.title('Draft Round vs. Fantasy Points')
plt.show()
```



In [23]:
```
# Scatter Plot: Draft Round vs. Total Yards
plt.figure(figsize=(6, 4))
plt.scatter(df['draft_round'], df['total_yards'])
plt.xlabel('Draft Round')
plt.ylabel('Total Yards')
plt.title('Draft Round vs. Total Yards')
plt.show()
```



In [24]:
```
# Pearson correlation
corr, _ = stats.pearsonr(df['draft_round'], df['fantasy_points_ppr'])
print(f"Pearson Correlation: {corr}")
```

```
Pearson Correlation: -0.4086080725726298
```

In [25]:
```
# Hypothesis Testing
first_round = df[df['round'] == 1]['fantasy_points_ppr']
later_round = df[df['round'] > 1]['fantasy_points_ppr']

t_stat, p_val = ttest_ind(first_round, later_round, equal_var=False)
print(f"T-Statistic: {t_stat}, P-Value: {p_val}")
```

```
T-Statistic: 18.39448593274092, P-Value: 1.7122973538443794e-67
```

In [26]:
```
# Regression Analysis
X = df[['round']]
y = df['fantasy_points_ppr']

X = sm.add_constant(X)  # Add intercept
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     fantasy_points_ppr   R-squared:                       0.047
Model:                            OLS   Adj. R-squared:                  0.046
Method:                 Least Squares   F-statistic:                     231.1
Date:                Mon, 24 Feb 2025   Prob (F-statistic):           5.46e-51
Time:                        20:08:59   Log-Likelihood:                -28290.
No. Observations:                4731   AIC:                         5.658e+04
Df Residuals:                    4729   BIC:                         5.660e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        134.8159      2.467     54.654      0.000     129.980     139.652
round        -10.2678      0.675    -15.201      0.000     -11.592      -8.944
==============================================================================
Omnibus:                      626.853   Durbin-Watson:                   0.793
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              900.745
Skew:                           1.018   Prob(JB):                    2.54e-196
Kurtosis:                       3.654   Cond. No.                         6.81
==============================================================================
```