Instructional Airplane Leasing Program

Overview

This dataset details every accident investigated by the National Transportation Safety Board over the last 40 years. This data details the make, models, weather conditions, flight phases, injuries, and deaths during each accident. The data could potentially help organizations find correlations, and identify conditions of an airplane flight that make it safer.

Key findings:

1. **Cessna** and **Piper** airplanes appear to offer better protection to its passengers during airplane accidents.
2. Accidents that occur during **taxi**, **takeoff**, and **landing** have the **lowest** fatalities rates. Accidents that occur during **maneuvers** and **cruise** phases have the **highest** fatality rates.
3. Flights occurring during the **summer** months tend to have **lower** fatality rates that flights occuring during the **winter** months.

Business Problem

In order to choose safe airplanes for the flight school, we will investigate these business questions:

1. How does the make of the aircraft affect airplane safety?
2. How does the phase of flight affect the survivability of airplane accidents?
3. How do the time of year and weather conditions affect the survival rate an airplane accident?

To measure airplane safety, we will consider the protection offered by the airplane during an accident. We will look at this three different ways

- **Survive:** The fraction of accidents with no fatalities.
- **Survival Rate:** The fraction of passengers killed during an accident
- **Injury Rate:** The fraction of passengers uninjured during an accident.

Data Understanding

Intake Data

In this analysis we read in the entire AviationData dataset. For each record, we use the following data:

- Event Date
- Make of Airplane
- Accidents injured, killed, minor injuries, and uninjured passengers in each accident
- Phase of Flight and weather conditions

In [31]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme("paper")

import scr.data_preparation as dp
import scr.data_visualizations as dv

df = pd.read_csv("Data/AviationDataClean.csv", dtype={"Latitude":object, "Lo
                                                      "Report.Status":object
```

```
/var/folders/ym/68nrz1n97wj0gz5413bhpqs80000gn/T/ipykernel_14652/1870390053.py:10: DtypeWarning: Columns (32) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv("Data/AviationDataClean.csv", dtype={"Latitude":object, "Longitude":object,
```

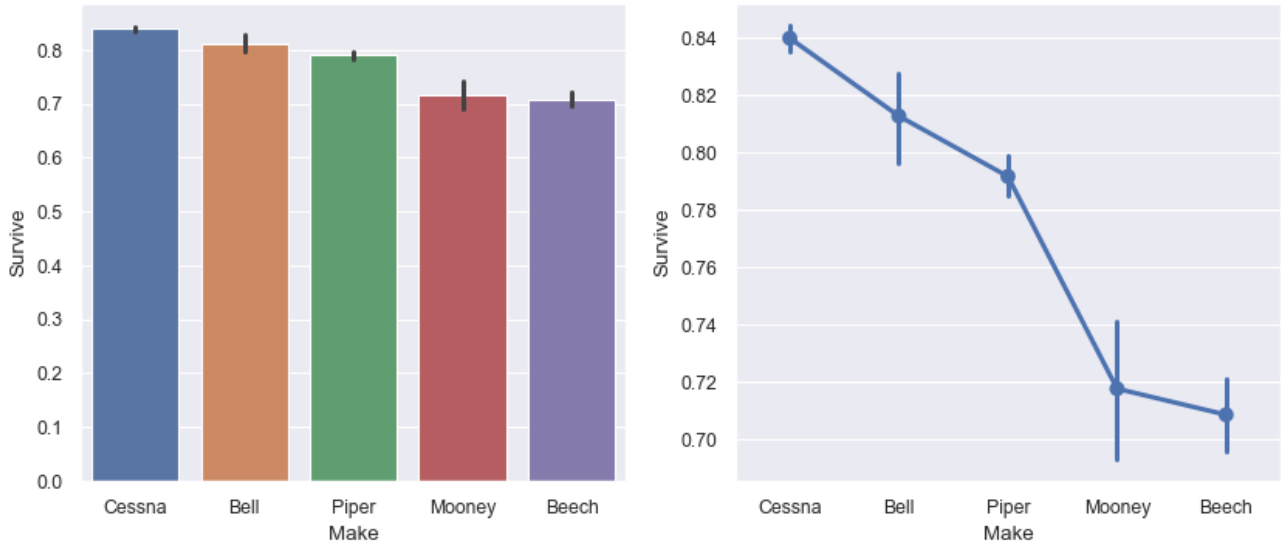
In [32]:
```
df = df.dropna(subset="Model")
df["Make_Model"] = df[["Make","Model"]].apply(lambda x: "-".join(x), axis=1)
```

Outcome Data

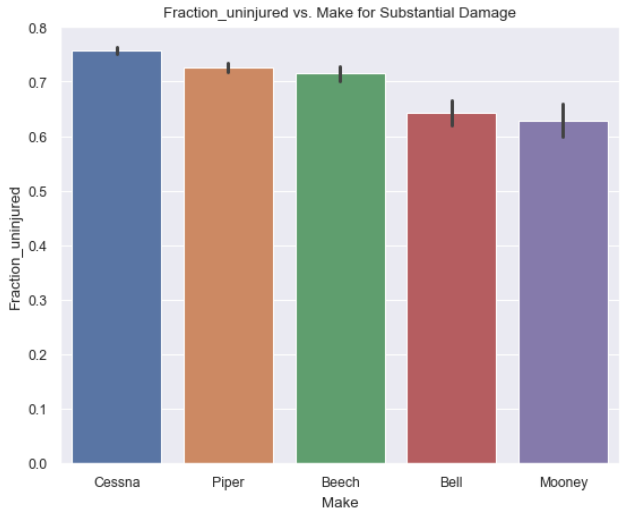
Analysis

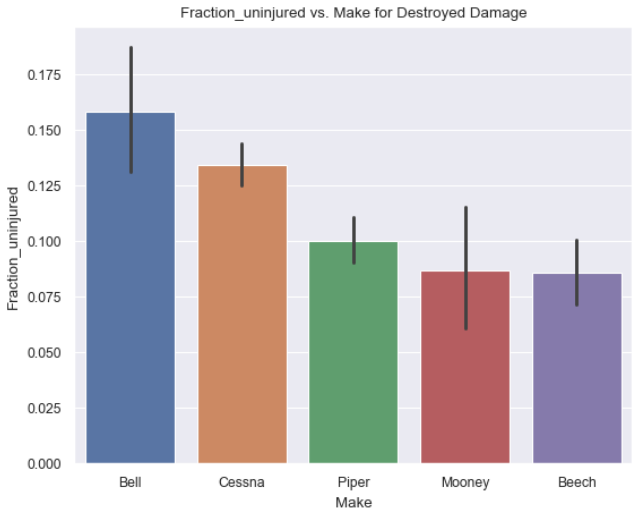
Airplane Safety by Make of Aircraft

In [33]:
```
fig, ax = plt.subplots(1,2, figsize = (10,4))

sns.barplot(data=df, x='Make', y = 'Survive', ax = ax[0], order=df[["Make","
sns.pointplot(data=df, x='Make', y = 'Survive', order=df[["Make","Survive"]]

fig.savefig("Images/SurviveMake")
```

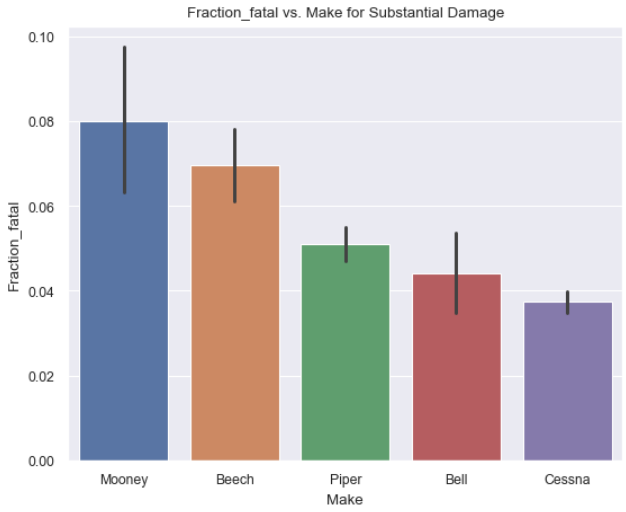


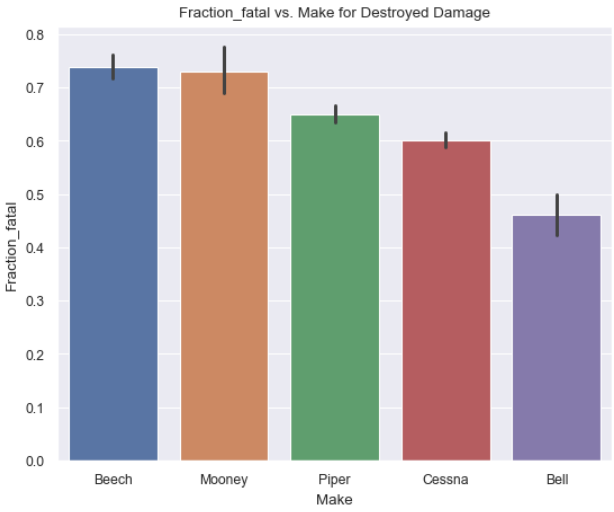
In [34]:
```
dv.performance_make_damage(df, "Fraction_uninjured")
```





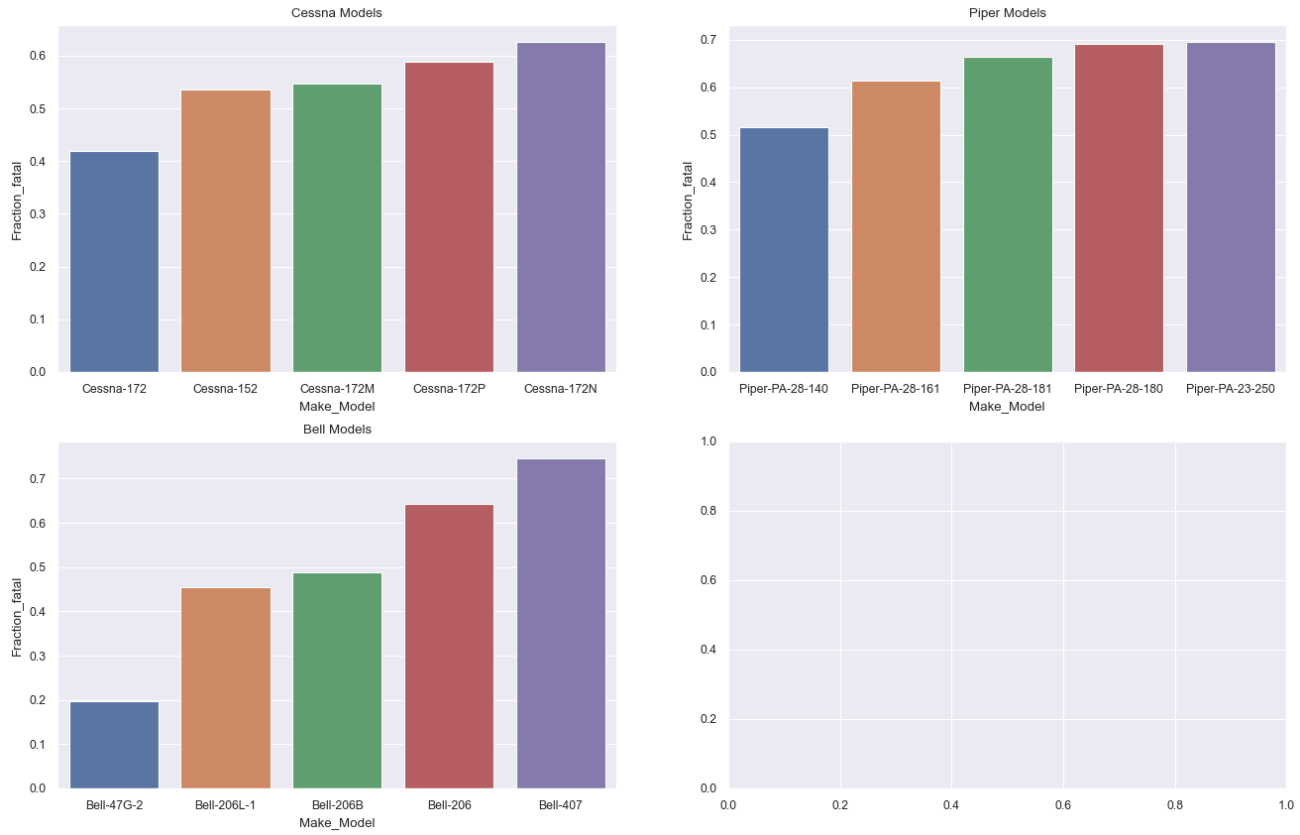
In [35]:
```
dv.performance_make_damage(df, "Fraction_fatal")
```



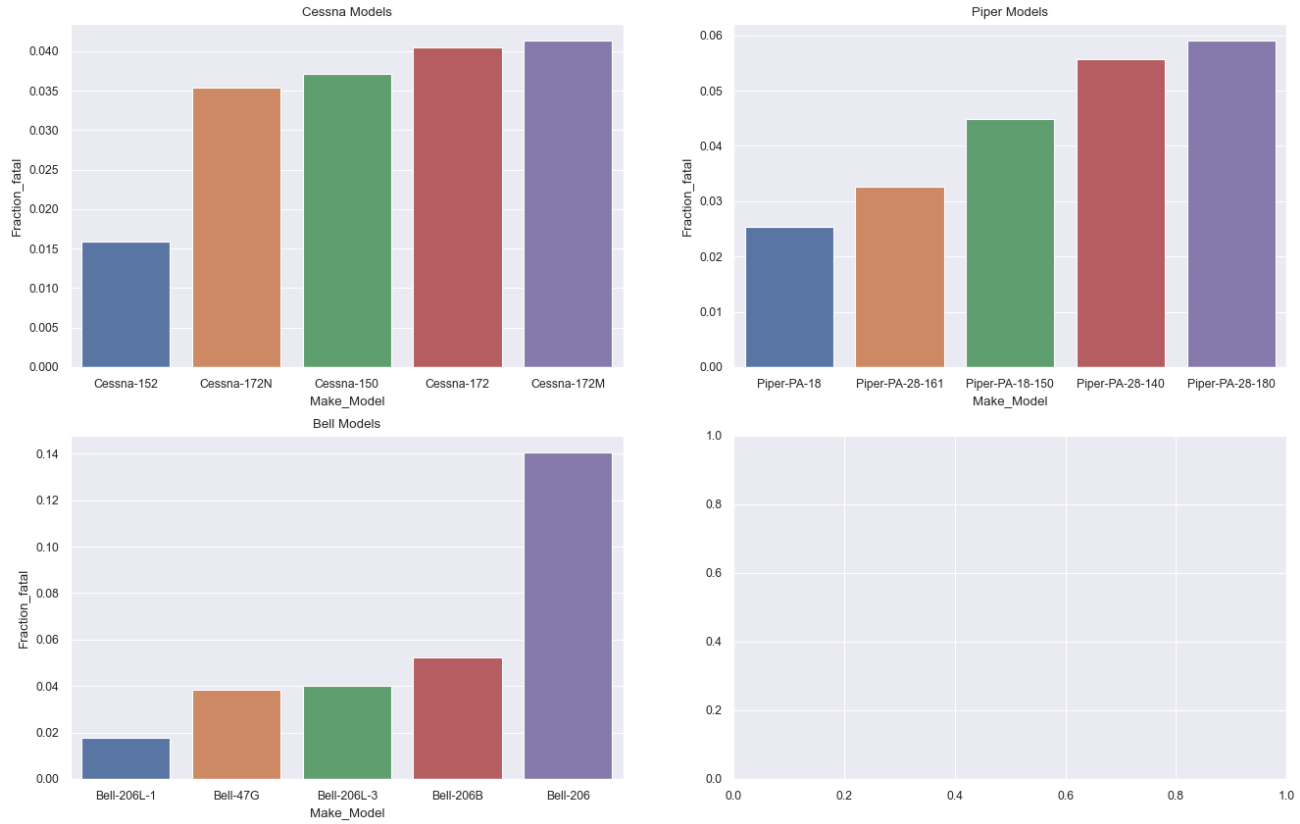


Model Safety

In [36]:
```
dfD = df[df['Aircraft.damage'] == 'Destroyed']
fig, ax = plt.subplots(2,2, figsize = (16,10))

fig.suptitle('Destroyed per Make-Model in Fatal', fontsize = 18)

dv.hi(dfD, 'Cessna', 'Fraction_fatal', ax[0][0])
dv.hi(dfD, 'Piper', 'Fraction_fatal', ax[0][1])
dv.hi(dfD, 'Bell', 'Fraction_fatal', ax[1][0])

fig.savefig("Images/ModelDestroyedFatal")
```



In [37]:
```
dfD = df[df['Aircraft.damage'] == 'Substantial']
fig, ax = plt.subplots(2,2, figsize = (16,10))

fig.suptitle('Substantial Damage per Make-Model in Fatal', fontsize = 18)

dv.hi(dfD, 'Cessna', 'Fraction_fatal', ax[0][0])
dv.hi(dfD, 'Piper', 'Fraction_fatal', ax[0][1])
dv.hi(dfD, 'Bell', 'Fraction_fatal', ax[1][0])

fig.savefig("Images/ModelSubstantialFatal")
```

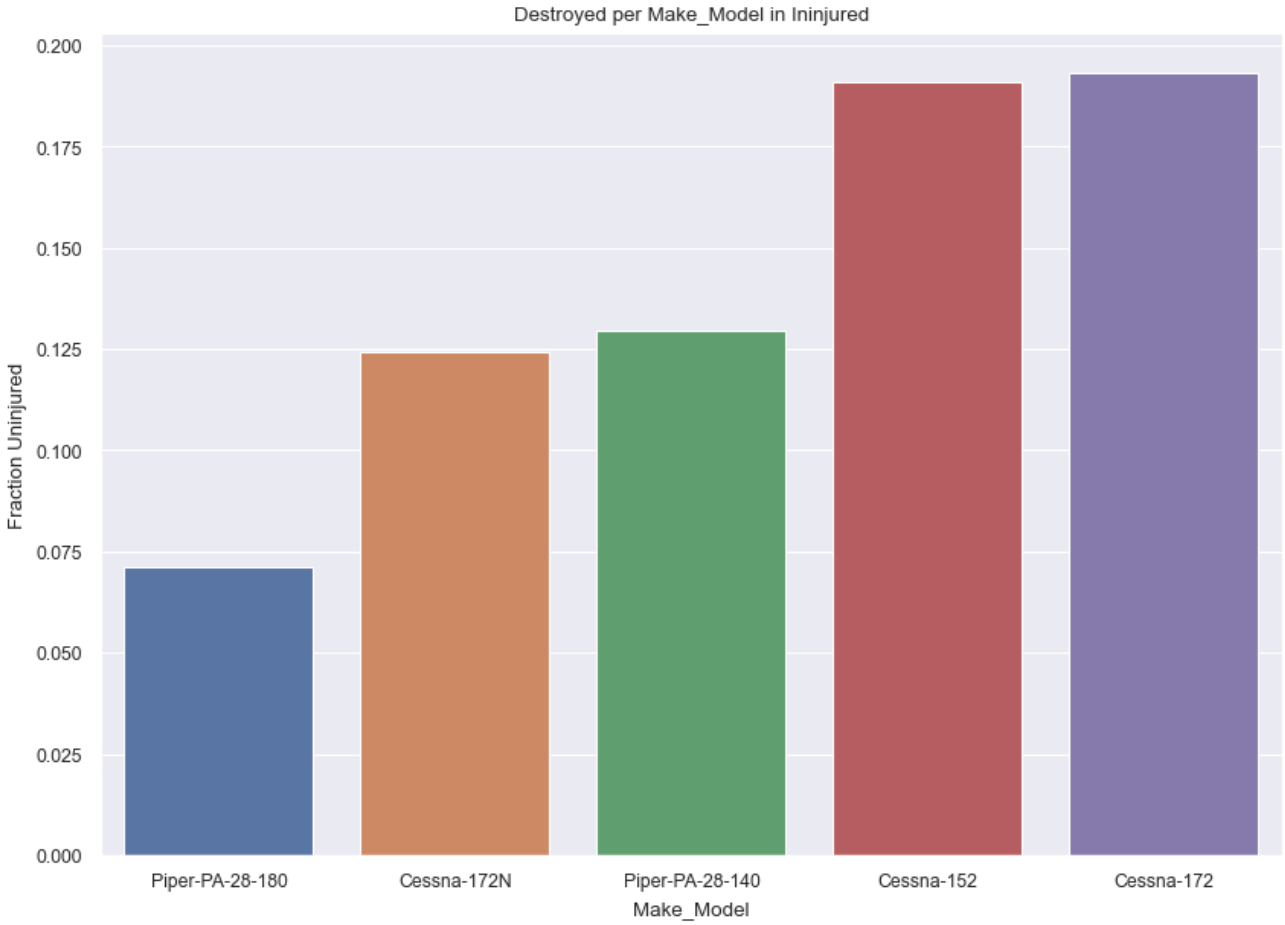

Substantial Damage per Make-Model in Fatal
Cessna Models
Fraction_fatal
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Cessna-152
Cessna-172N
Cessna-150
Cessna-172
Cessna-172M
Make_Model
Piper Models
Fraction_fatal
0.06
0.05
0.04
0.03
0.02
0.01
0.00
Piper-PA-18
Piper-PA-28-161
Piper-PA-18-150
Piper-PA-28-140
Piper-PA-28-180
Make_Model
Bell Models
Fraction_fatal
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
Bell-206L-1
Bell-47G
Bell-206L-3
Bell-206B
Bell-206
Make_Model
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

In [38]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 48596 entries, 0 to 49245
Data columns (total 39 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   Unnamed: 0              48596 non-null  float64
 1   Event.Id                48596 non-null  object 
 2   Investigation.Type      48596 non-null  object 
 3   Accident.Number         48596 non-null  object 
 4   Event.Date              48596 non-null  object 
 5   Location                48585 non-null  object 
 6   Country                 48457 non-null  object 
 7   Latitude                16406 non-null  object 
 8   Longitude               16405 non-null  object 
 9   Airport.Code            29887 non-null  object 
 10  Airport.Name            31425 non-null  object 
 11  Injury.Severity         48567 non-null  object 
 12  Aircraft.damage         48401 non-null  object 
 13  Aircraft.Category       14782 non-null  object 
 14  Registration.Number     48581 non-null  object 
 15  Make                    48596 non-null  object 
 16  Model                   48596 non-null  object 
 17  Amateur.Built           48586 non-null  object 
 18  Number.of.Engines       48596 non-null  float64
 19  Engine.Type             47532 non-null  object 
 20  FAR.Description         14759 non-null  object 
 21  Schedule                3944 non-null   object 
 22  Purpose.of.flight       47649 non-null  object 
 23  Air.carrier             6923 non-null   object 
 24  Total.Fatal.Injuries    48596 non-null  float64
 25  Total.Serious.Injuries  48596 non-null  float64
 26  Total.Minor.Injuries    48596 non-null  float64
 27  Total.Uninjured         48596 non-null  float64
 28  Weather.Condition       48059 non-null  object 
 29  Broad.phase.of.flight   38102 non-null  object 
 30  Report.Status           47022 non-null  object 
 31  Publication.Date        39539 non-null  object 
 32  Survive                 47950 non-null  object 
 33  total.passengers        47950 non-null  float64
 34  Month                   47950 non-null  float64
 35  Year                    47950 non-null  float64
 36  Fraction_fatal          47916 non-null  float64
 37  Fraction_uninjured      47916 non-null  float64
 38  Make_Model              48596 non-null  object 
dtypes: float64(11), object(28)
memory usage: 14.8+ MB
```

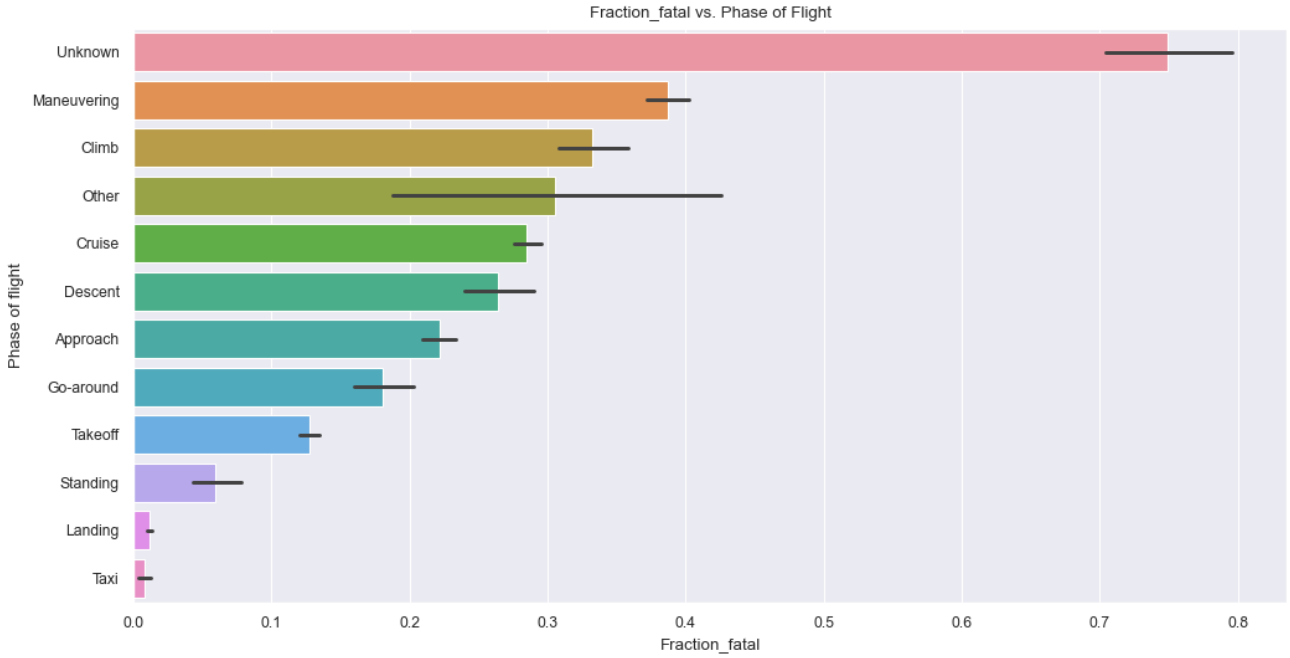
In [40]:
```
dv.best_models(df)
```

```
/Users/jgoett/Library/CloudStorage/OneDrive-Personal/Documents/Fall2024/Flatiron-Live/Phase1/Project/Phase1-Airplane-Zach-Jeff/scr/data_visualizations.py:58: FutureWarning: Calling float on a single element Series is deprecated and will raise a TypeError in the future. Use float(ser.iloc[0]) instead
  xy = pd.DataFrame([(x,float(ave_frac.loc[x])) for x in top_5]).sort_values(by=1)
```

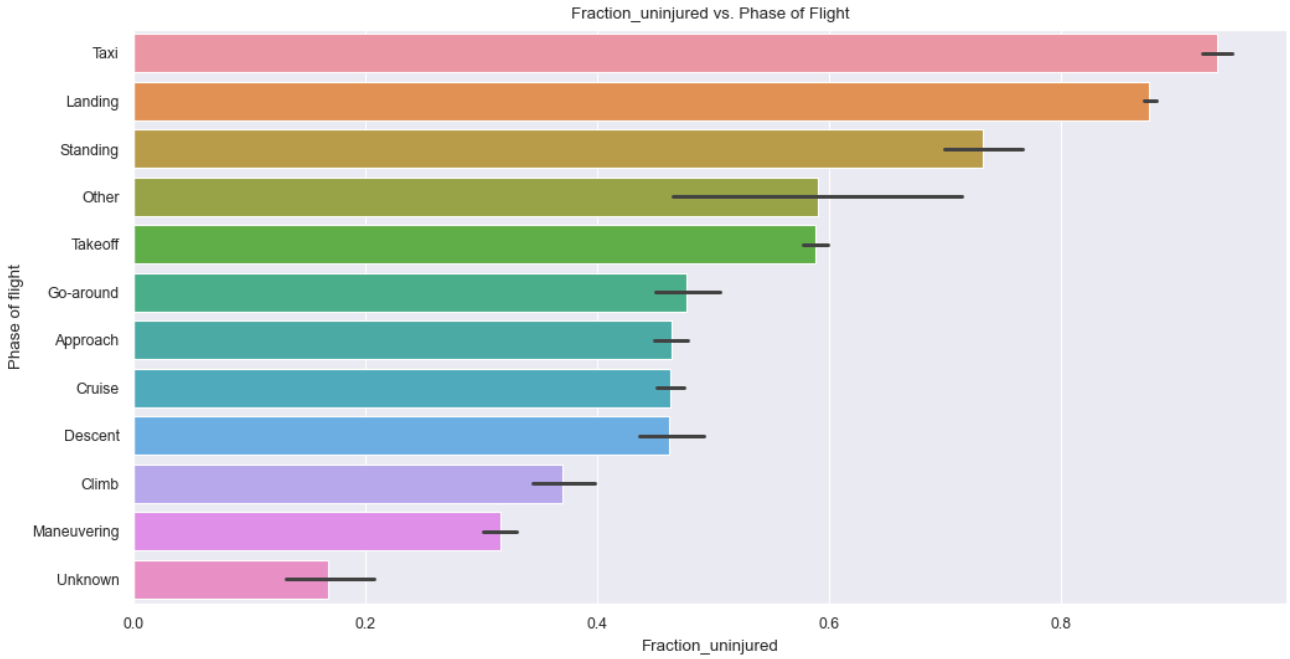


Airplane Safety during Phases of Flight

In [41]:
```
dv.performance_phase(df, "Fraction_fatal")
```



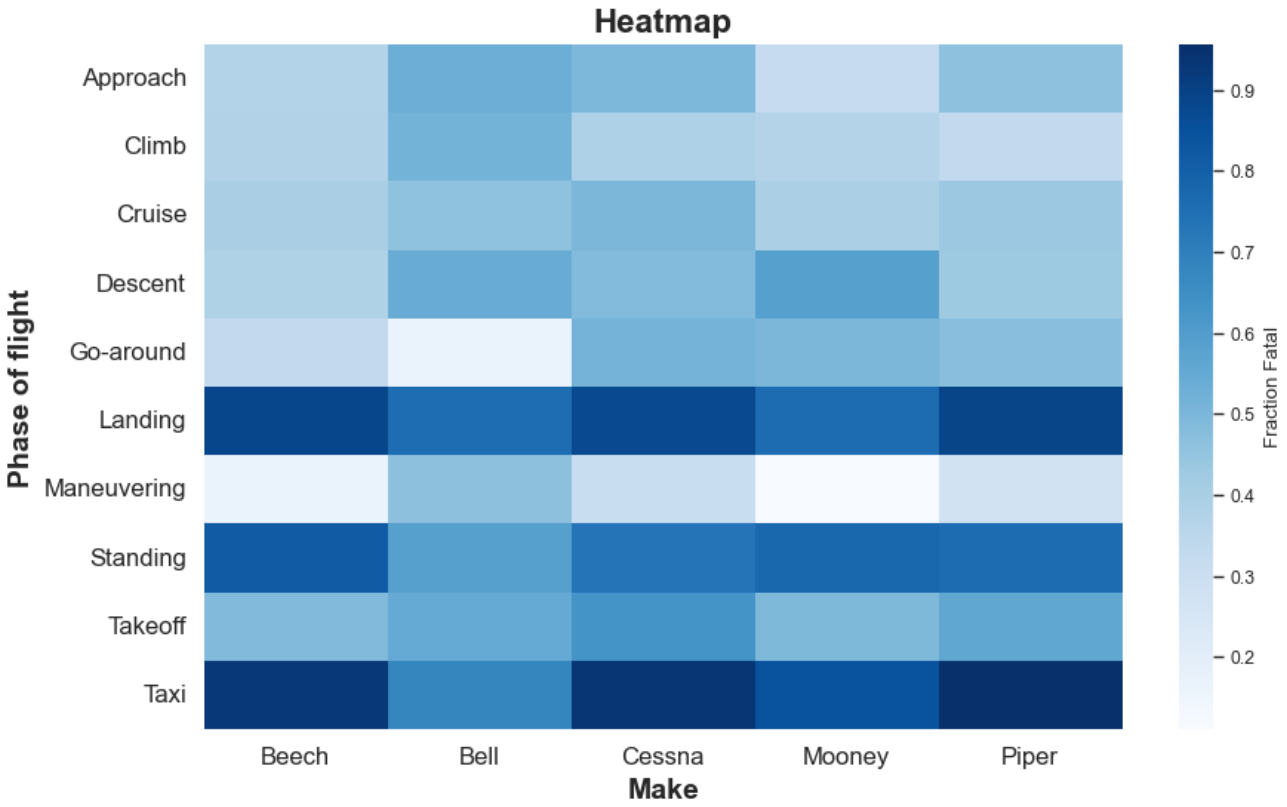
In [42]:
```
dv.performance_phase(df, "Fraction_uninjured")
```



Calculating Heatmap

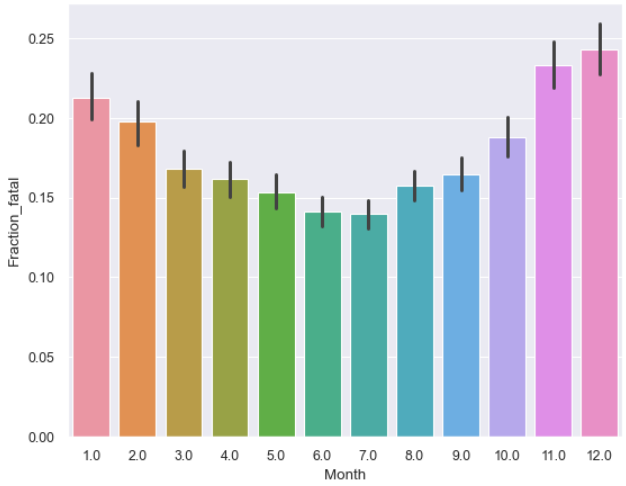
In [43]:
```
phases_flight = df["Broad.phase.of.flight"].value_counts().index[:-2]
df_phases = df[df["Broad.phase.of.flight"].map(lambda x: x in phases_flight)
```

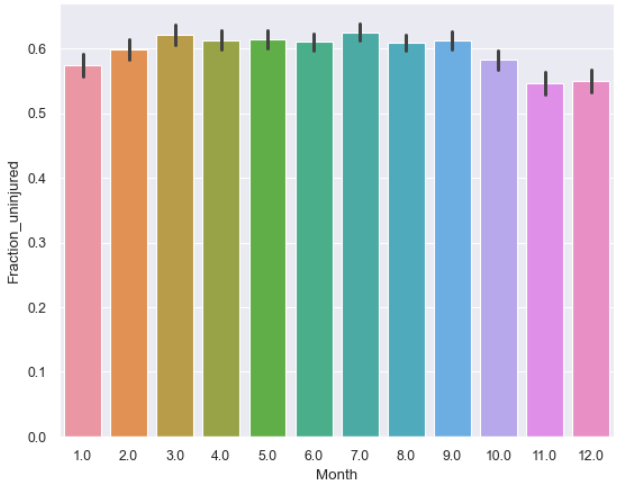
In [44]:
```
fig, ax = plt.subplots(figsize=(10,6))


table = pd.pivot_table(df_phases, index=['Make'],columns=['Broad.phase.of.fl
sns.heatmap(table, cmap="Blues", cbar_kws={"label":"Fraction Fatal"}, ax=ax)
ax.set_title('Heatmap', fontsize=16, fontweight = 'bold')
ax.tick_params(labelsize=12)
ax.set_xlabel('Make', fontweight = 'bold', fontsize=14)
ax.set_ylabel('Phase of flight', fontweight = 'bold', fontsize=14)

fig.savefig("Images/Uninjured_Make_Phases", bbox_inches = 'tight')
```



Airplane Safety during Different Months of the Year

In [45]:
```
fig, ax = plt.subplots(1,2,figsize=[14,5])

sns.barplot(data=df, x='Month', y = 'Fraction_fatal', ax=ax[0])
sns.barplot(data=df, x='Month', y = 'Fraction_uninjured', ax=ax[1])

fig.savefig("Images/Survival_month")
```





Conclusions

- Consider **Cessna** or **Piper** makes (Cessna 172, 158, or Piper PA-28)
- Focus on safety during **climb**, **cruize**, and **maneuvering** phases of flight
- Focus on safety during **winter** flying.

Next Steps

- Investigate the geopraphical locations of accidents to gain insight.
- Investigate weather conditions.