# Part I

## Data Exploration

1. Load the hflights dataset and call it hflights.

Hflight <- hflights::hflights

1. Explain in **words** what each feature. Hint: use ?
2. Show some basic statistics (Hint: use glimpse)
3. How many row and columns?
4. What is each column type? (numeric, character, date etc.)

## Data Manipulation

### Select and Mutate

1. Print out a **table** containing only `*UniqueCarrier`* and columns ending with 'Delay'
2. Select all columns from `*relevant\_cols`*

***relevant\_cols*** = c("Year","Month","Day","ArrTime")

1. Use the lookup table `**lut**` (From the appendix) to translate the *UniqueCarrier* column of *hflights* and call the new column UniqueCarrier\_name
2. Change CancellationCode from "" to "E"
   1. Show **table** of CancellationCode
3. Columns Year, Month and DayofMonth represent a date. Create a new column *Flight\_Date* of class Date.
   1. Test that the new column is of **class** date
4. Bring the new column to the front and place it after *DayofMonth*.
   1. Use `names(hflight)` to test
5. Add the new variable ***ActualGroundTime*** to a copy of *hflights* and save the result.

***ActualGroundTime*** = *ActualElapsedTime*-*airtime*

1. Add the new variable ***AverageSpeed*** to a copy of *hflights* and save the result.

***AverageSpeed*** = *Distance* / *AirTime* \* 60

### Handle missing values

1. Show missing values distribution
2. Replace NA values of *ActualElapsedTime* with the mean time for all flights grouped by *Origin* and *Dest*

### Filtering Data

1. Filter All flights that traveled 500 miles or more
2. Filter *UniqueCarrier\_name* in {'jetblue', 'Southwest', 'Delta'}
3. Filter all flights that departed before 5am or arrived after 10pm

(*DepTime* < 500 or *ArrTime* > 2200)

1. Filter all rows where both *DepTime* **and** *ArrTime* are missing

### Arranging Data

1. Arrange *hflight* by departure delays (*Depdelay*)
2. Arrange *hflight* by *UniqueCarrier* and **descending** *Depdelay*

### Group\_by, Summarise and Aggregate functions

1. Find minimum and maximum Distance for each *UniqueCarrier*
2. Find min, max, mean, and median for all numeric columns except Year, Month and *DayofMonth* grouped by Origin and Destination. Hint: use across

### Join

1. Create a data frame **numeric\_hflights\_stats** which comprises of min, max, mean, and median for all numeric columns group by Origin and Destination
2. Create a data frame **character\_hflights\_stats** which comprises of first, last and n\_distinctfor all character columns group by Origin and Destination
3. Join **numeric\_hflights\_stats** and **character\_hflights\_stats** by the appropriate key

## Data Analysis

1. Check how many planes each company (*UniqueCarrier*) has
2. Calculate the average landing delays (*ArrDelay*)
3. Check how many flights were canceled due to the weather
4. What is the percentage of flights that were canceled out of the total number of flights to the destination ‘DAL’ for the company
5. Present the distribution of cancellations and the reasons for cancellation for flights over over 800 km. Sort the results in descending order.
6. How many destinations does the company reach, and what is the most common destination with the largest number of flights.
7. Print the 5 most cancelled aircrafts (*TailNum*)
8. What destinations (*Dest*) has the longest delay (*DepDelay*)
9. Is the average speed in winter (Month= 1,2,3) slower than the average speed in summer (Month= 6,7,8).

Hint: Create a new variable Season= Winter,Summer,None.

Filter Season is not None

group by season

Calculate avg\_speed=mean(Distance/AirTime,na.rm=T)

## Data visualization

### Plot analysis (ggplot)

1. Load the **diamonds** dataset and sample 1000 random observation
2. Plot `*price*` against `*carat*` and color each `*cut*` differently
3. Plot a single linear regression fit on the data above. Color the smoothed line red (geom\_smooth(method='lm', formula= y~x))
4. Add labels to the figure
   1. Title: "Diamond price per carat"
   2. X-label: "Carat Size"
   3. Y-label: "Price$"
5. Make facet by color and clarity

### Distribution analysis

1. Plot histogram of carat
2. Plot histogram for carat for each cut (on the same plot)
3. Plot density for each cut (on the same plot)

# Part II

Choose a **classification** data set and use your R skill to analyze it and find a set of feature rules that have a good separation. Be as creative as possible!

A classification data set is where one feature is treated as a target, in our hflight dataset for example we can consider the Canceled feature to be the target.

See example.

# Appendix

Lut <- c("AA" = "American", "AS" = "Alaska", "B6" = "jetblue", "CO" = "Continental",

"DL" = "Delta", "OO" = "skywest", "UA" = "United", "US" = "US\_Airways",

"WN" = "Southwest", "EV" = "Atlantic\_Southeast", "F9" = "Frontier",

"FL" = "airtran", "MQ" = "American\_Eagle", "XE" = "expressjet", "YV" = "Mesa")

**Aggregate functions defined in R:**

Min(x) - minimum value of vector x max(x) - maximum value of vector x

Mean(x) - mean value of vector x median(x) - median value of vector x

Quantile(x, p) - pth quantile of vector x sd(x) - standard deviation of

Vector x var(x) - variance of vector x IQR(x) - Inter Quartile Range

(IQR) of vector x diff(range(x)) - total range of vector x

**Dplyr has some of its own aggregate functions**

First(x) - The first element of vector x

Last(x) - The last element of vector x nth(x, n) -The nth element of vector x

N() - The number of rows in the data.frame or group of observations that summarise() describes

N\_distinct(x) – The number of unique values in vector x

Good website: https://r4ds.had.co.nz/

Datasets from: UCI repository

<https://archive.ics.uci.edu/ml/datasets.php?format=&task=cla&att=&area=&numAtt=&numIns=&type=&sort=nameUp&view=table>

Kaggle

<https://www.kaggle.com/datasets>