Exam 1

Anish Bhurtyal

Academic Honesty Statement

I, Anish Bhurtyal, hereby state that I have not communicated with or gained information in any way from my classmates or anyone other than the Professor during this exam, and that all work is my own.

Load packages

```
# load required packages here

library(dplyr)
library(tidyr)
library(nycflights13)
library(readxl)
library(ggplot2)
```

Logistics

Add your code and/or narrative in the spaces below each question. Add code chunks as needed. Use as many lines as you need, but keep your narrative concise.

Packages

In addition to `tidyverse`, you will need the `nycflights13` package for the data. You will first need to install these packages and then load them. Feel free to use any other library.

The Data

1. The `nycflights13` package contains information about all flights that departed from NYC (e.g. EWR, JFK and LGA) in 2013. The main data is in the `flights` data frame, but there are additional data sets which may help understand what causes delays, specifically:

- weather: hourly meteorological data for each airport
- planes: construction information about each plane
- airports: airport names and locations
- airlines: translation between two letter carrier codes and names

2. The dataset `oslo-biomarkers.xlsx` contains data from a medical study about patients with disc herniations, performed at the Oslo University Hospital, Ullevål. Blood samples were collected from a number of patients with disc herniations at three time points: 0 weeks (first visit at the hospital), 6 weeks and 12 months. The levels of some biomarkers related to inflammation were measured in each blood sample.

3. The dataset `oslo-covariates.xlsx` contains information about the patients, such as age, gender and smoking habits.

Questions

Question 1

a.What does it mean for data analysis to be "reproducible"?

Data analysis is meant to be reproducible if it has the capacity to produce consistent computational outputs by adopting the same input data, computational steps, techniques, code, and analysis conditions. Reproducible refers to the practice of making available all the data, software sources, and tools needed to reproduce the findings reported in a research paper.

The main objective of reproducible data analysis is to guarantee computational reproducibility, i.e., the capability for another researcher to utilize one's code and data to independently produce identical results. Reproducible analysis ensures accuracy or consistency of the outcomes. proper documentation of the process gathering, preprocessing, and analyzing the data makes the data analysis reproducible

b. Discuss the toolkit for reproducibility.

Automation — Scripts can be used to automate these activities.R enables you to document your actions so that you can repeat your analysis and share it to others.

Literate programming (code, narrative, output in one place)—like using rmarkdown files and writing narration, comments

Using version control—– Keeping track of code changes and using some sort of git repository like Github. allows us to track code changes and history

Question 2

a. What is Exploratory data analysis (EDA)?

Data analysis via visual methods is called exploratory data analysis (EDA). With the use of statistical summaries and graphical representations, it is used to identify trends, patterns, or to verify assumptions.

We employ visual methods like graphs, charts, and other types of visualization. This is due to the fact that when trends and anomalies are depicted graphically, our natural pattern-detection abilities make it much simpler for us to identify them. An easy identification of outliers is also possible with EDA

b. Why do we visualize data? Explain using Anscombe's quartet as a case study.

identify patterns, trends and outliers in large data sets. identify the correlations between the relationship of x-y variables find trends over time understanding about reach and reports of company sales, targets, achievement scores and so on

Anscombe's quartet consists of four datasets with virtually similar simple statistical properties that, when graphed, appear vastly different. Hence we will be able to see such anomaly only when we graphed. Thus we do data visualization to notice trends and clusters and any unusual patterns or outliers

Question 3

a. Write a function called `fahr_to_celsius` that converts temperatures from Fahrenheit to Celsius. Print out the results when the temperature is 32^oF.

```
fahr_to_celsius<- function(n) {
 (n-32)*5/9
  #(9/5) * n + 32;
}
fahr_to_celsius(32)
```

```
## [1] 0
```

32 fahr is 0 celc

b. Write another function called `celsius_to_kelvin` that converts Celsius into Kelvin. Print out the results when the temperature is 0^oC.

```
celsius_to_kelvin<- function(n) {
 n+273.15
  #(9/5) * n + 32;
}
celsius_to_kelvin(0)
```

```
## [1] 273.15
```

0 celc is 273.15 kel

c. Compose the two functions in Parts 3a. and 3b. into a new function called `fahr_to_kelvin` that converts Fahrenheit to Kelvin. Print out the results when the temperature is 32^oF.

```
fahr_to_kelvin<- function(n) {
  celsius_to_kelvin(fahr_to_celsius(n))

  #celcius = fahr_to_celsius(n)
  #kelvin = celsius_to_kelvin(celcius)
  #kelvin
}
fahr_to_kelvin(32)
```

```
## [1] 273.15
```

32 fahr is 273.156 kel

** For Questions 4,5, and 6, use the `nycflights13` package **

Question 4

What are the ten most common destinations for flights from NYC airports in 2013? Make a table that lists these in descending order of frequency and shows the number of flight heading to each airport.

```
flightsdata <- flights

table(flightsdata['dest'])%>%
  as.data.frame() %>%
  arrange(desc(Freq)) %>%
  slice(1:10)
```

```
##    dest  Freq
## 1   ORD 17283
## 2   ATL 17215
## 3   LAX 16174
## 4   BOS 15508
## 5   MCO 14082
## 6   CLT 14064
## 7   SFO 13331
## 8   FLL 12055
## 9   MIA 11728
## 10  DCA  9705
```

ORD,ATL,LAX, BOS,MCO are the top 10 most common destinations for flights from NYC airports in 2013. frequency and shows the number of flight heading to each airport is shown in the above table

Question 5

Which airlines have the most flights departing from NYC airports in 2013? Make a table that lists these in descending order of frequency and shows the number of flights for each airline. In your narrative mention the names of the airlines as well. Hint: You can use the `airlines` dataset to look up the airline name based on carrier code.

```
top_airlines <- table(flightsdata['carrier'])%>%
  as.data.frame() %>%
  arrange(desc(Freq))
top_airlines
```

```
##    carrier  Freq
## 1       UA 58665
## 2       B6 54635
## 3       EV 54173
## 4       DL 48110
## 5       AA 32729
## 6       MQ 26397
## 7       US 20536
## 8       9E 18460
## 9       WN 12275
## 10      VX  5162
## 11      FL  3260
## 12      AS   714
## 13      F9   685
## 14      YV   601
## 15      HA   342
## 16      OO    32
```

```
airlinesdata <- airlines

#using full join to get the airline names based on carrier code by use of `airlines` dataset
df= top_airlines %>% full_join(airlinesdata,by="carrier")
df
```

```
##    carrier  Freq                        name
## 1       UA 58665       United Air Lines Inc.
## 2       B6 54635             JetBlue Airways
## 3       EV 54173    ExpressJet Airlines Inc.
## 4       DL 48110        Delta Air Lines Inc.
## 5       AA 32729      American Airlines Inc.
## 6       MQ 26397                   Envoy Air
## 7       US 20536             US Airways Inc.
## 8       9E 18460           Endeavor Air Inc.
## 9       WN 12275      Southwest Airlines Co.
## 10      VX  5162              Virgin America
## 11      FL  3260 AirTran Airways Corporation
## 12      AS   714        Alaska Airlines Inc.
## 13      F9   685      Frontier Airlines Inc.
## 14      YV   601          Mesa Airlines Inc.
## 15      HA   342      Hawaiian Airlines Inc.
## 16      OO    32       SkyWest Airlines Inc.
```

"United Air Lines Inc." "JetBlue Airways" "ExpressJet Airlines Inc." "Delta Air Lines Inc." (in descending order) are the top 4 most common airlines(others in table above) for flights from NYC airports in 2013

Question 6

Consider only flights departing from New York on 1 January that year.

a. Are there missing data?

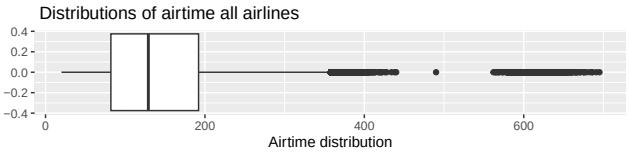
```
on_jan_1 <- flightsdata%>%
  filter(flightsdata$year == 2013 & flightsdata$month == 1 & flightsdata$day ==1 )
sum(is.na(on_jan_1))
```

```
## [1] 35
```

The dataset has 35 missing data

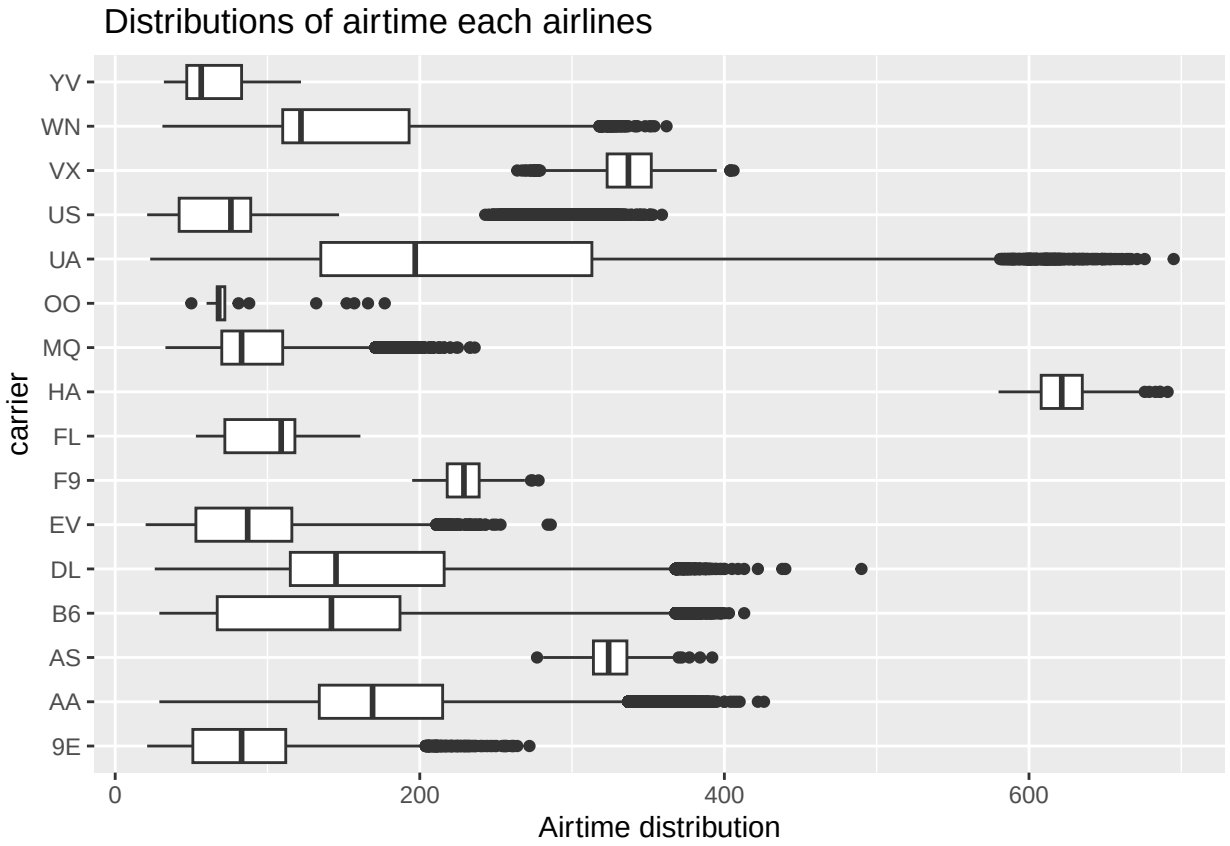
b. Are there any differences between the different carriers? Focusing on delays and the amount of time spent in the air.

```
flightsdata %>%
  filter(!is.na(air_time)) %>%
  ggplot(aes(x = air_time)) +
  geom_boxplot() +
  labs(title = " Distributions of airtime all airlines",
       x = "Airtime distribution")
```



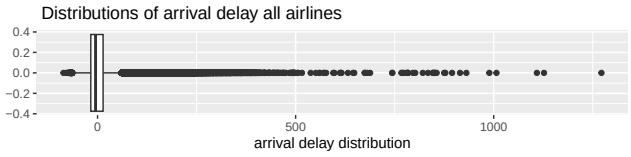
The outliers seem to be airtime > 350

```
ggplot(flightsdata, aes(x = air_time, y = carrier)) +
  geom_boxplot()+
  labs(title = " Distributions of airtime each airlines",
       x = "Airtime distribution")
```

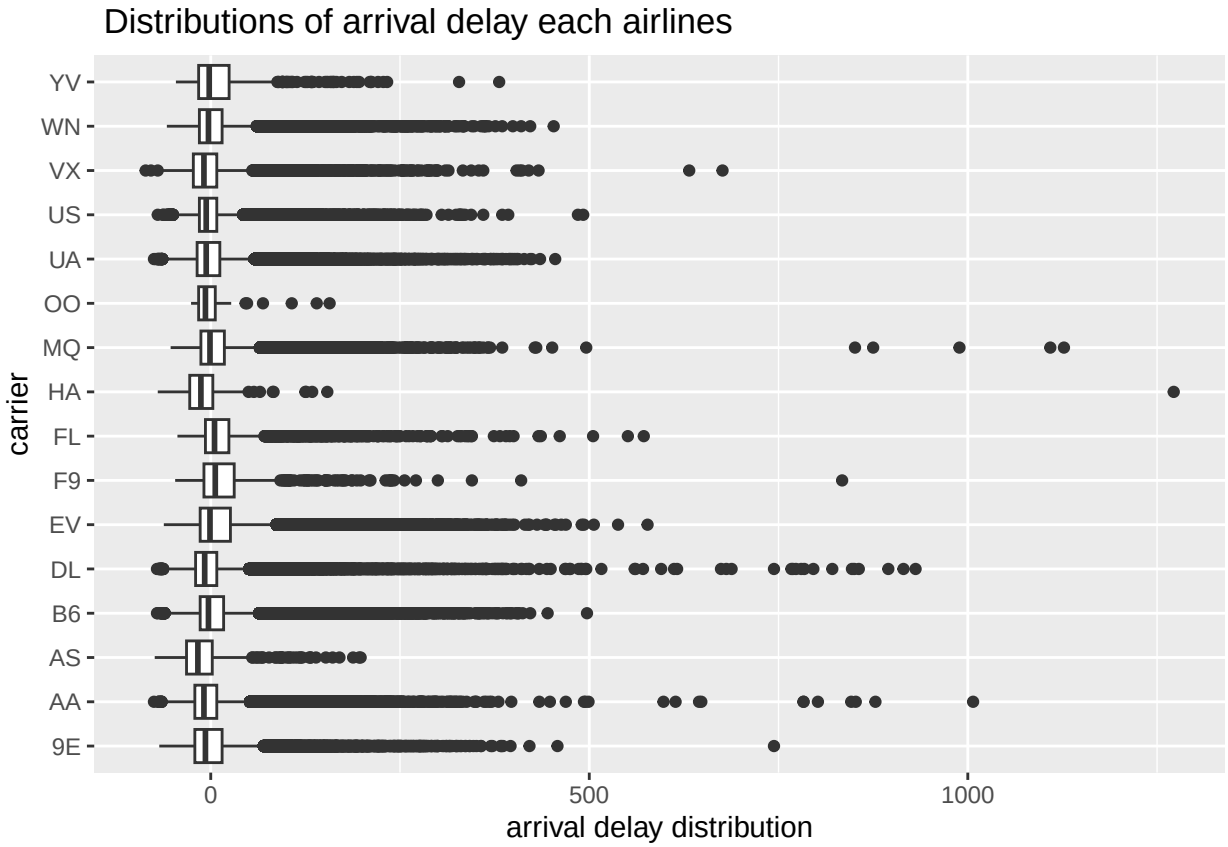


There are certainly many differences in the distribution of airtimes in various airlines as depicted in the boxplot above. reason may be that they have different flight journey schedules.

```
flightsdata %>%
  filter(!is.na(arr_delay)) %>%
  ggplot(aes(x = arr_delay)) +
  geom_boxplot() +
  labs(title = " Distributions of arrival delay all airlines",
       x = "arrival delay distribution")
```




```
ggplot(flightsdata, aes(x = arr_delay, y = carrier)) +
  geom_boxplot()+
  labs(title = " Distributions of arrival delay each airlines",
       x = "arrival delay distribution")
```



Here's the distribution of arr delay in various airlines as depicted in the boxplot above. Arrival delay is common in arlines and can happen due to bad weather, airport traffic and among all these flights the delay seem to be similarly distributed across all airlines

c. Are there any outliers? Focusing on delays and the amount of time spent in the air.

From last figure in b, we saw the outliers seem to be airtime > 350. HA airline entire flight time is an outlier in this observaton There are outliers in arrival time dealy also as shown in the box plot above

Question 7

Load the `oslo-biomarkers.xlsx data`. Then do the following using dplyr/tidyr:

```
oslo_data <- read_excel("oslo-biomarkers.xlsx")
oslo_data
```

```
## # A tibble: 347 x 10
##    PatientID.t~1 `IL-8` VEGF-~2   OPG TGF-b~3 `IL-6` CXCL9 CXCL1 `IL-18` `CSF-1`
##    <chr>          <dbl>   <dbl> <dbl>   <dbl>  <dbl> <dbl> <dbl>   <dbl>   <dbl>
##  1 126-0weeks      7.64    11.5  10.2    8.85   3.53  6.19  9.5     7.91    8.42
##  2 126-6weeks      7.09    11.6  10.4    8.86   3.89  6.1   9.07    7.92    8.41
##  3 127-0weeks      6.92    10.9  10.3    6.61   2.69  6.17  7.28    7.98    8.41
##  4 127-6weeks      7.18    11.6  10.4    8.57   2.59  6.35  8.6     7.93    8.55
##  5 127-12months    6.89    11.2  10.2    7.46   3.96  6.17  8.8     7.97    8.46
##  6 128-0weeks      8.57    12.4  10.6    8.49   3.71  7.34  9.9     8.72    8.73
##  7 128-6weeks      6.94    11.6  10.6    7.44   3.86  7.16  8.57    8.63    8.51
##  8 128-12months    6.49    11.0  10.1    6.41   4.62  7.98  8.17    8.68    8.56
##  9 129-0weeks      8.16    11.1  10.6    8.81   3.84  5.82  9.17    7.51    8.43
## 10 129-6weeks      6.52    10.7  10.2    6.8    2.96  6.16  6.71    7.28    8.14
## # ... with 337 more rows, and abbreviated variable names
```

```
## #   1: PatientID.timepoint, 2: `VEGF-A`, 3: `TGF-beta-1`
```

a. Split the `PatientID.timepoint` column in two parts: one with the patient ID and one with the timepoint.

```
splitted_oslo_data <- oslo_data %>%
  separate(PatientID.timepoint, c('PatientID', 'TimePoint'))

splitted_oslo_data
```

```
## # A tibble: 347 x 11
##    PatientID TimePoint `IL-8` `VEGF-A`   OPG TGF-be~1 `IL-6` CXCL9 CXCL1 `IL-18`
##    <chr>     <chr>      <dbl>    <dbl> <dbl>    <dbl>  <dbl> <dbl> <dbl>   <dbl>
##  1 126       0weeks      7.64     11.5  10.2     8.85   3.53  6.19  9.5     7.91
##  2 126       6weeks      7.09     11.6  10.4     8.86   3.89  6.1   9.07    7.92
##  3 127       0weeks      6.92     10.9  10.3     6.61   2.69  6.17  7.28    7.98
##  4 127       6weeks      7.18     11.6  10.4     8.57   2.59  6.35  8.6     7.93
##  5 127       12months    6.89     11.2  10.2     7.46   3.96  6.17  8.8     7.97
##  6 128       0weeks      8.57     12.4  10.6     8.49   3.71  7.34  9.9     8.72
##  7 128       6weeks      6.94     11.6  10.6     7.44   3.86  7.16  8.57    8.63
##  8 128       12months    6.49     11.0  10.1     6.41   4.62  7.98  8.17    8.68
##  9 129       0weeks      8.16     11.1  10.6     8.81   3.84  5.82  9.17    7.51
## 10 129       6weeks      6.52     10.7  10.2     6.8    2.96  6.16  6.71    7.28
## # ... with 337 more rows, 1 more variable: `CSF-1` <dbl>, and abbreviated
## #   variable name 1: `TGF-beta-1`
```

`PatientID.timepoint` column Splitted in two parts in table above

b. Sort the table by patient ID, in numeric order.

```
  typeof(splitted_oslo_data$PatientID)
```

```
## [1] "character"
```

The data type of Patient id is char so we would need to convert it to int to do numeric sorting

```
 splitted_oslo_data$PatientID <- as.integer(splitted_oslo_data$PatientID)
 typeof(splitted_oslo_data$PatientID)
```

```
## [1] "integer"
```

```
splitted_oslo_data %>%
  arrange(PatientID)
```

```
## # A tibble: 347 x 11
##    PatientID TimePoint `IL-8` `VEGF-A`   OPG TGF-be~1 `IL-6` CXCL9 CXCL1 `IL-18`
##        <int> <chr>      <dbl>    <dbl> <dbl>    <dbl>  <dbl> <dbl> <dbl>   <dbl>
##  1         1 0weeks      8.16     12.3  10.5     8.68   2.6   6.55  9.61    8.52
##  2         1 6weeks      7.11     11.6  10.3     7.57   2.28  7.83  9.47    8.39
##  3         1 12months    8.6      12.5  10.7     8.5    2.57  6.68  9.62    8.79
##  4         3 0weeks      6.56     11.2  10.5     6.85   2.55  5.37  6.69    7.7
##  5         3 6weeks      6.33     11.2  10.6     6.63   2.26  5.51  6.67    7.82
##  6         3 12months    7.44     11.7  10.7     7.82   2.73  5.53  8.78    7.77
##  7         4 0weeks      6.42     11.1  10.8     6.95   5.64  5.44  7.74    8
##  8         4 6weeks      7.71     11.5  11.0     7.74   5.14  5.63  8.32    8.15
##  9         4 12months    7.24     11.5  10.9     7.37   4.33  5.72  8.28    8.21
## 10         5 0weeks      6.42     11.2  10.7     7.26   1.92  5.29  7.14    7.68
## # ... with 337 more rows, 1 more variable: `CSF-1` <dbl>, and abbreviated
## #   variable name 1: `TGF-beta-1`
```

The above table has sorted data by patient ID, in numeric order

c. Reformat the data from long to wide, keeping only the `IL-8` and `VEGF-A` measurements.

```
wide_oslo_data <- splitted_oslo_data %>%
  pivot_wider(
    id_cols = PatientID,
    names_from = TimePoint,
    values_from = c("IL-8", "VEGF-A")
  )

wide_oslo_data
```

```
## # A tibble: 118 x 7
##    PatientID `IL-8_0weeks` `IL-8_6weeks` `IL-8_12months` VEGF-~1 VEGF-~2 VEGF-~3
##        <int>         <dbl>         <dbl>           <dbl>   <dbl>   <dbl>   <dbl>
##  1       126          7.64          7.09           NA       11.5    11.6    NA
##  2       127          6.92          7.18            6.89    10.9    11.6    11.2
##  3       128          8.57          6.94            6.49    12.4    11.6    11.0
##  4       129          8.16          6.52            6.52    11.1    10.7    10.9
##  5       130          8.84          7.22            7.22    12.5    11.3    11.2
##  6       131          7.14          7.77            6.2     10.9    11.5    10.4
##  7       132          6.71          5.59            5.81    10.9    10.5    10.3
##  8       133          7.32          8.13            6.33    10.5    12.4    10.8
##  9       134          7.91          6.79            7.31    11.8    11.0    11.2
## 10       135          8.26          8.19            6.57    13.0    12.7    11.1
## # ... with 108 more rows, and abbreviated variable names 1: `VEGF-A_0weeks`,
## #   2: `VEGF-A_6weeks`, 3: `VEGF-A_12months`
```

Here we can see that the long form has gone to wide form in terms of IL-8 and VEGA-A column attribute values

d. Merge the wide data frame from part c. with the `oslo-covariates.xlsx` data, using patient ID as key.

```
oslo_covar_data <- read_excel("oslo-covariates.xlsx")

merged_data <- wide_oslo_data %>%
  left_join(oslo_covar_data,
            by = "PatientID")

merged_data
```

```
## # A tibble: 118 x 12
##    Patie~1 IL-8_~2 IL-8_~3 IL-8_~4 VEGF-~5 VEGF-~6 VEGF-~7   Age Sex (~8 Smoke~9
##      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>
##  1     126    7.64    7.09   NA       11.5    11.6    NA      41       2       1
##  2     127    6.92    7.18    6.89    10.9    11.6    11.2    35       1       2
##  3     128    8.57    6.94    6.49    12.4    11.6    11.0    57       2       2
##  4     129    8.16    6.52    6.52    11.1    10.7    10.9    28       1       2
##  5     130    8.84    7.22    7.22    12.5    11.3    11.2    58       2       1
##  6     131    7.14    7.77    6.2     10.9    11.5    10.4    35       2       2
##  7     132    6.71    5.59    5.81    10.9    10.5    10.3    42       2       2
##  8     133    7.32    8.13    6.33    10.5    12.4    10.8    42       2       1
##  9     134    7.91    6.79    7.31    11.8    11.0    11.2    31       2       2
## 10     135    8.26    8.19    6.57    13.0    12.7    11.1    43       2       1
## # ... with 108 more rows, 2 more variables: `VAS-at-inclusion` <dbl>,
## #   `Vas-12months` <dbl>, and abbreviated variable names 1: PatientID,
```

```
## #   2: `IL-8_0weeks`, 3: `IL-8_6weeks`, 4: `IL-8_12months`, 5: `VEGF-A_0weeks`,
## #   6: `VEGF-A_6weeks`, 7: `VEGF-A_12months`, 8: `Sex (1=male, 2=female)`,
## #   9: `Smoker (1=yes, 2=no)`
```

Performed left join to see attributes of both tables in one table

e. Use the `oslo-covariates.xlsx` data to select data for smokers from the wide data frame in part c.

```
#renaming 4th column for easiness
colnames(oslo_covar_data)[4] <- "smoking"
oslo_covar_data$smoking <- as.integer(oslo_covar_data$smoking)

only_smoker_data <- filter(oslo_covar_data, smoking ==2)
only_smoker_data
```

```
## # A tibble: 79 x 6
##    PatientID   Age `Sex (1=male, 2=female)` smoking `VAS-at-inclusion` Vas-12m~1
##        <dbl> <dbl>                    <dbl>   <int>               <dbl>     <dbl>
##  1         1    55                        2       2                 3         4
##  2         3    32                        1       2                 7.2       0.5
##  3         4    33                        2       2                 2.7       0.5
##  4         6    39                        1       2                 3.5       5
##  5         7    38                        2       2                 7         3
##  6         8    48                        2       2                 7         5
##  7        14    41                        1       2                 2         4
##  8        16    34                        2       2                 6         1.4
##  9        17    32                        2       2                 4         3.5
## 10        21    46                        2       2                 7         2
## # ... with 69 more rows, and abbreviated variable name 1: `Vas-12months`
```

```
merged_data %>% semi_join(only_smoker_data,
                          by = "PatientID")
```

```
## # A tibble: 79 x 12
##    Patie~1 IL-8_~2 IL-8_~3 IL-8_~4 VEGF-~5 VEGF-~6 VEGF-~7   Age Sex (~8 Smoke~9
##      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <dbl>   <dbl>   <dbl>
##  1     127    6.92    7.18    6.89    10.9    11.6    11.2    35       1       2
##  2     128    8.57    6.94    6.49    12.4    11.6    11.0    57       2       2
##  3     129    8.16    6.52    6.52    11.1    10.7    10.9    28       1       2
##  4     131    7.14    7.77    6.2     10.9    11.5    10.4    35       2       2
##  5     132    6.71    5.59    5.81    10.9    10.5    10.3    42       2       2
##  6     134    7.91    6.79    7.31    11.8    11.0    11.2    31       2       2
##  7     136    7.4     6.97    6.14    11.6    11.2    10.6    26       1       2
##  8     138    8.22    8.11    6.61    12.5    12.4    11.0    49       1       2
##  9     139    6.75    7.6     5.96    11.2    11.6    10.5    31       2       2
## 10     141    6.4     9.7     5.85    11.1    12.5    10.4    32       1       2
## # ... with 69 more rows, 2 more variables: `VAS-at-inclusion` <dbl>,
## #   `Vas-12months` <dbl>, and abbreviated variable names 1: PatientID,
## #   2: `IL-8_0weeks`, 3: `IL-8_6weeks`, 4: `IL-8_12months`, 5: `VEGF-A_0weeks`,
## #   6: `VEGF-A_6weeks`, 7: `VEGF-A_12months`, 8: `Sex (1=male, 2=female)`,
## #   9: `Smoker (1=yes, 2=no)`
```

Semi join either returns each row from input A, or it does not. No row duplication can occur. Regular join duplicates rows if there are multiple matches on the join predicate

Referred sites

1) https://poldrack.github.io/psych-open-science-guide/4_reproducibleanalysis.html

2) DATA 101: Intro to Data Science classs pptx Osei
3) https://www.geeksforgeeks.org/what-is-exploratory-data-analysis/ 4)https://sqlperformance.com/2018/02/sql-plan/row-goals-part-2-semi-joins