Causal discovery demo

Reminders you can delete if you don't want them in your code or report:

Shortcuts:

- Use **CTRL ALT I** to add a new code chunk
  - Use a new code chunk to answer a new question (or subquestion)
- Use **ALT -** for the assignment operator <-
- Use **CTRL SHIFT M** for the pipe function %>%
- Check out the dropdown menu at top right next to "Run" by clicking on the down arrow
  - It will show you the shortcuts to **run a line, run current chunk, all chunks above, and all chunks**

Good advice on function:

- Use **read_csv** (NOT read.csv) to load the data
- When **%>%** function cannot be found, load the tidyverse library again
- Avoid loading the libraries you won't use frequently. Just refer to them directly.
  - **To load packages**: Run library("package_name")
  - **Use without loading**: Specify package as **package_name::function_name**
  - **To install packages:** Run install.packages("package_name") or use RStudio menu
    - You might want to install packages for your project. Otherwise, you are covered.
- R/RStudio is **case sensitive**, so lower vs. Upper case are different

Good advice on style:

- We are following the **Tidyverse Style Guide** (https://style.tidyverse.org/), **so does Google** (https://google.github.io/styleguide/Rguide.html)
- Name objects and columns/variables by…
  - either using an underscore such as **weekly_sales (preferable)**
  - or starting lowercase and using uppercase for each word such as **weeklySales (still readable)**
- You will likely see a mix of the two styles in the labs/assignments
  - Recently, I use **the former style for object names and the latter for column/variable names**
- There is much more to style: Keep up with the spaces, correct indentations, etc.
  - When in doubt, visit **style.tidyverse.org** and/or **use the Styler package**

The following is your first chunk to start with. Remember, you can add chunks using the menu above (Insert -> R) or using the keyboard shortcut Ctrl+Alt+I. A good practice is to use different code chunks to answer different questions (& subquestions). You can delete this comment if you like.

```r
# Load the Airbnb dataset

dfa <- read_csv('data/airbnb-project-msba-sampled-10k.csv')
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 153995 Columns: 100
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (51): listing_url, state, city, name, summary, space, description, pict...
## dbl  (32): id, high_booking, host_id, latitude, longitude, accommodates, bat...
## lgl  (13): host_is_superhost, is_location_exact, requires_license, host_has_...
## date  (4): date, host_since, first_review, last_review
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Replicating Lab 2

#Creating treatment column
df<-dfa%>%
mutate(treatment = ifelse(date>="2019-11-01",1,0))%>%
relocate(treatment,date)

#Data cleaning
df$price = as.numeric(gsub("\\$", "", df$price))
```

```
## Warning: NAs introduced by coercion
```

```r
df1<-df%>%
  select(id,bedrooms,accommodates,minimum_nights,review_scores_rating,room_type,price,treatment,high_boo

df1=drop_na(df1)
```

```r
# Estimate the model using feols without fixed effect
cf <- feols(high_booking ~  -1 + treatment + price + accommodates + bedrooms
            + minimum_nights + review_scores_rating +
              factor(room_type), data = df1)

cf$coeftable[1,]
```

```
##             Estimate  Std. Error t value     Pr(>|t|)
## treatment 0.05077591 0.003066011 16.5609 1.655389e-61
```

```r
#Estimating the model using feols with fixed effect
cf2 <- feols(high_booking ~  -1 + treatment + price + accommodates + bedrooms
             + minimum_nights + review_scores_rating +
               factor(room_type)|id, data = df1)
```

```
## The variable 'factor(room_type)Shared room' has been removed because of collinearity (see $collin.va
```

```r
cf2$coeftable[1,]
```

```
##            Estimate  Std. Error  t value      Pr(>|t|)
## treatment 0.0879689 0.002424142 36.28868 4.525355e-286
```

```r
#Estimating the model using feols with fixed effect as id+city+state
cf3 <- feols(high_booking ~  -1 + treatment + price + accommodates + bedrooms
             + minimum_nights + review_scores_rating +
               factor(room_type)|id+factor(city)+factor(state), data = df1)

cf3$coeftable[1,]
```

```
##             Estimate  Std. Error  t value     Pr(>|t|)
## treatment 0.08797062 0.002425082 36.27531 7.31653e-286
```

```r
#Estimating the model using feols without fixed effect on price
cf4 <- feols(price ~  -1 + treatment + high_booking + accommodates + bedrooms
             + minimum_nights + review_scores_rating +
               factor(room_type), data = df1)

cf4$coeftable[1,]
```

```
##            Estimate Std. Error   t value     Pr(>|t|)
## treatment -4.706111   0.786169 -5.986132 2.157182e-09
```

```r
#Estimating the model using feols with fixed effect as id on price
cf5 <- feols(price ~  -1 + treatment + high_booking + accommodates + bedrooms
             + minimum_nights + review_scores_rating +
               factor(room_type)|id, data = df1)
```

```
## The variable 'factor(room_type)Shared room' has been removed because of collinearity (see $collin.va
```

```r
cf5$coeftable[1,]
```

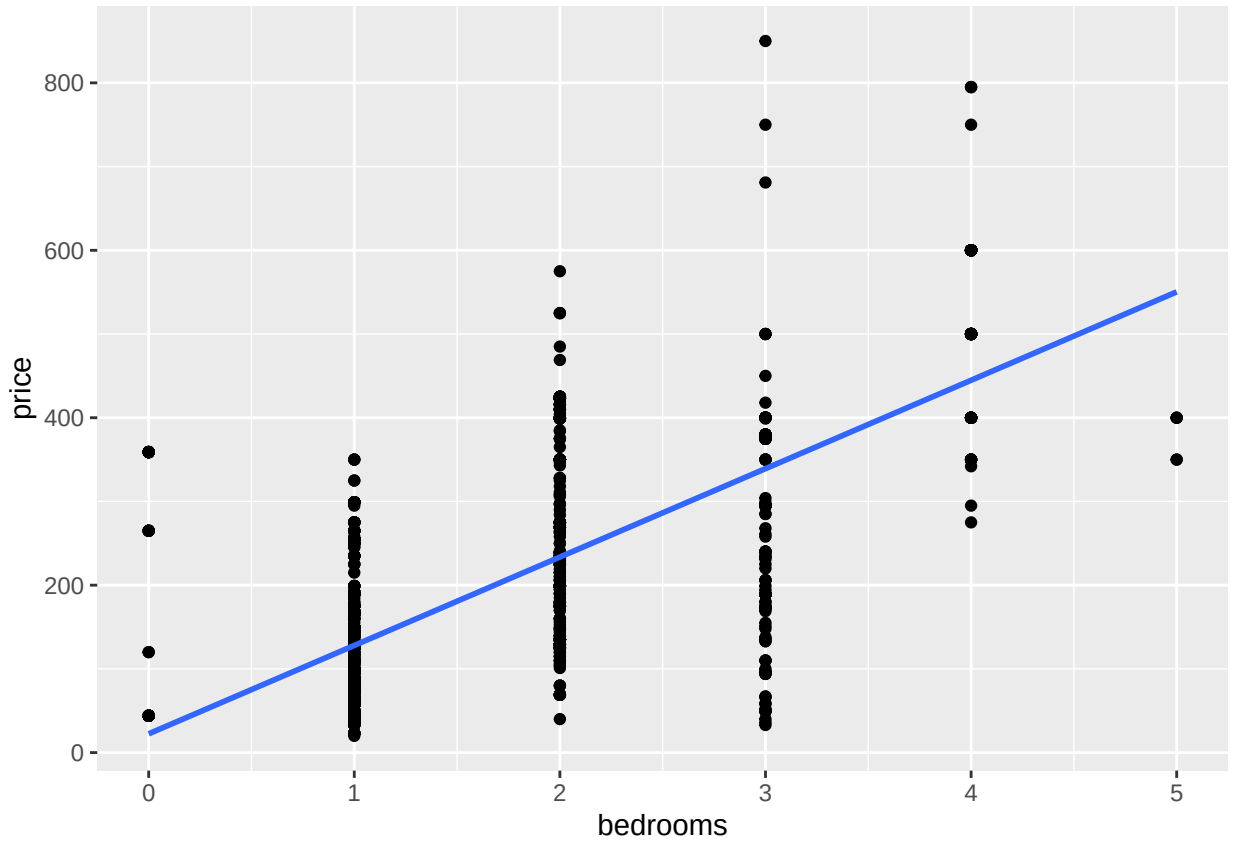
```
##             Estimate Std. Error  t value   Pr(>|t|)
## treatment -0.5939765  0.3503789 -1.69524 0.09003361
```

```r
#sampling data
unique_values <- df1%>%
  distinct(id)
set.seed(3.14159)
sampled_values <- unique_values %>%
  pull() %>%
  sample(100)

sampled_data <- df1%>%
  filter(id %in% sampled_values)

#plotting without group by id
ggplot(sampled_data, aes(x = bedrooms, y = price)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

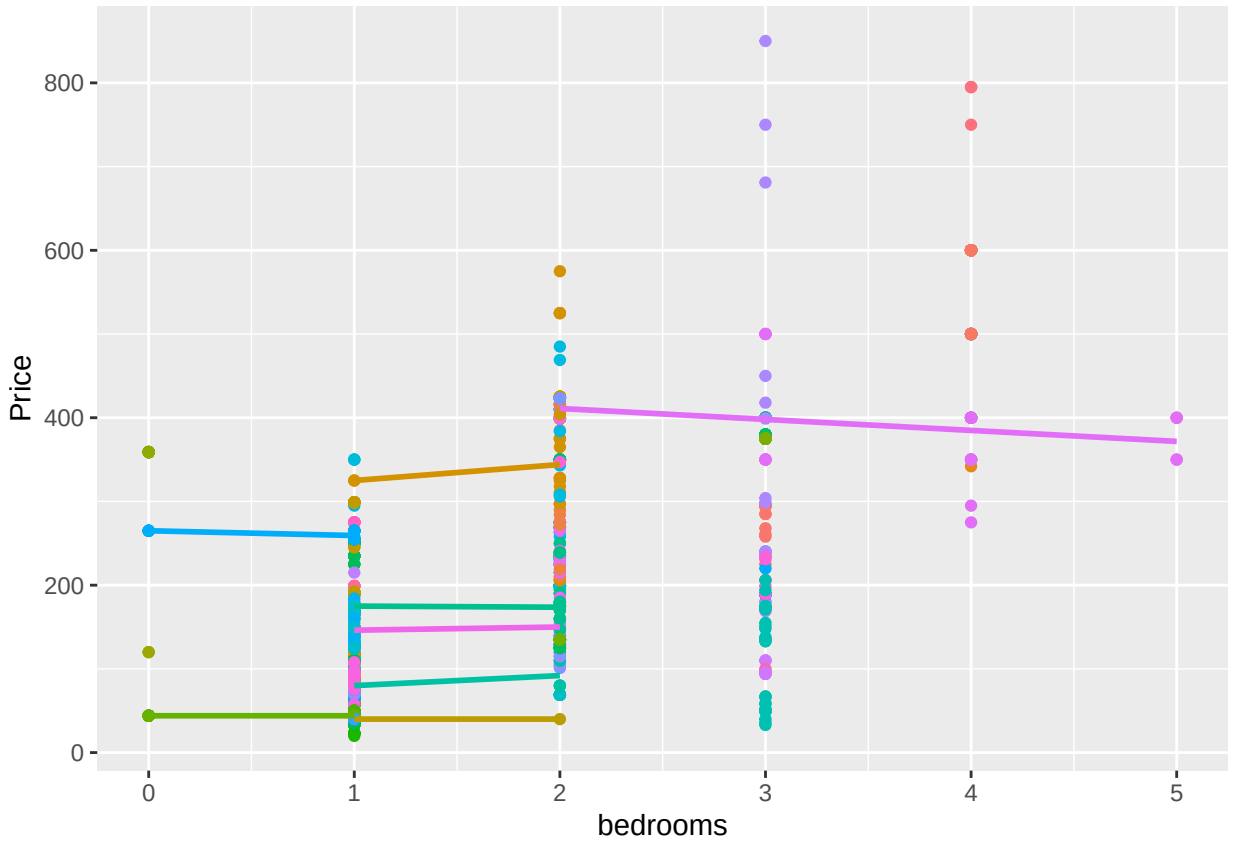
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
#plotting with group by id
ggplot(sampled_data, aes(bedrooms, price, color = factor(id))) +
  geom_point() +
  labs(x = "bedrooms", y = "Price", color = "id") +

  # Add trend line for each cluster
  stat_smooth(method = "lm", se = FALSE)+
  theme(legend.position = "none")
```

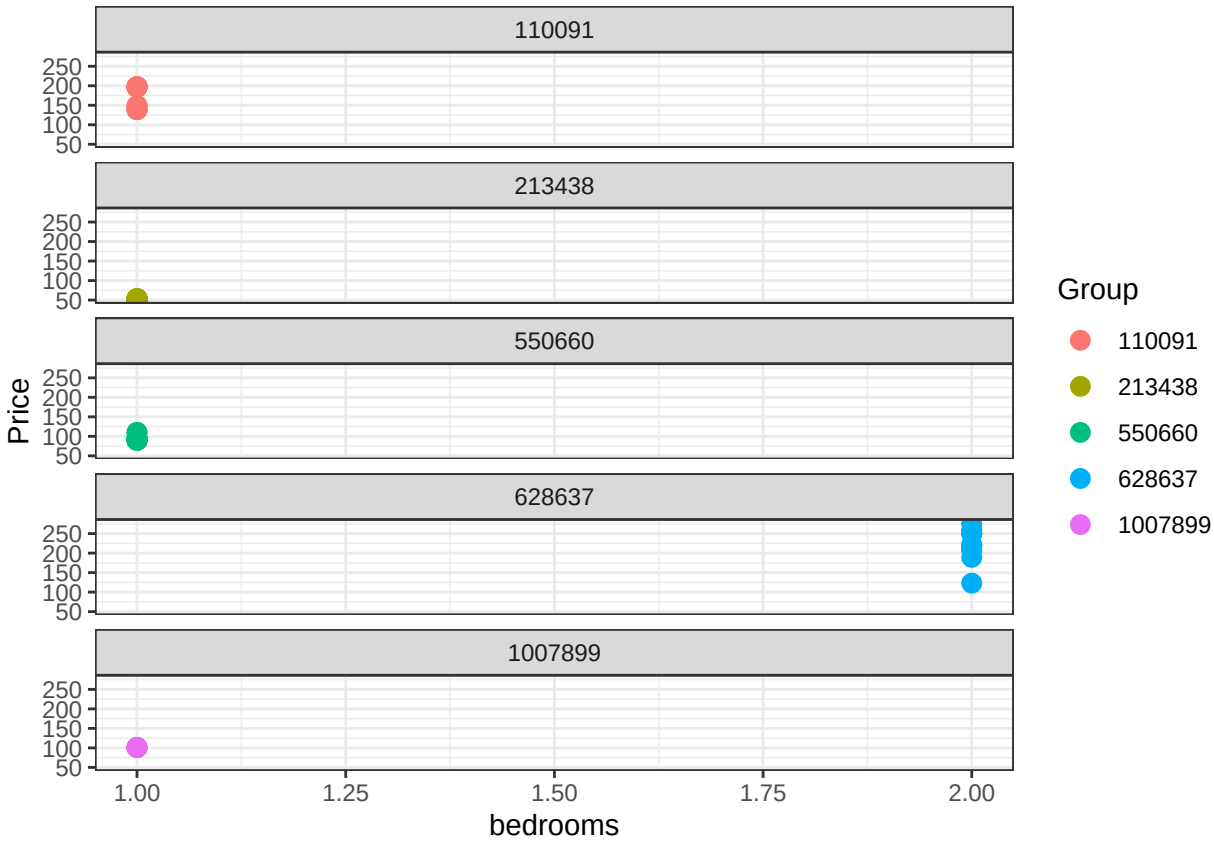
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
set.seed(2.14159)
sampled_values <- unique_values %>%
  pull() %>%
  sample(5)

sampled_data <- df1%>%
  filter(id %in% sampled_values)

ggplot(sampled_data, aes(x = bedrooms, y = price)) +
  geom_point(aes(color = factor(id)), size = 3) +
  facet_wrap(~id, ncol = 1) +
  labs(x = "bedrooms", y = "Price") +
  scale_color_discrete(name = "Group") +
  theme_bw()
```


110091
213438
550660
628637
1007899
250
200
150
100
50
Price
1.00
1.25
1.50
1.75
2.00
bedrooms
Group
110091
213438
550660
628637
1007899