

Solution code to postponed exam ECON3170/4170 Fall 2022

I. The world's billionaires

```
library(tidyverse)

setwd("D:/Dropbox_(UiO)/Undervisning/ECON4170/exam/e22uts")

# a)
forbes <- read_csv("forbes_2022_billionaires.csv")
summary(forbes$age)
max(forbes$finalWorth)

forbes %>%
  filter(rank<=100) %>%
  summarise(mean=mean(finalWorth))

# b)
countries <- forbes %>%
  group_by(country) %>%
  summarise(n=n()) %>%
  arrange(desc(n))

countries %>%
  top_n(12) %>%
  ggplot(aes(x=reorder(country,-n),y=n)) +
  geom_col()

# c)
forbes %>%
  filter(rank<=200) %>%
  ggplot(aes(x=rank,y=finalWorth)) +
  geom_line()

forbes %>%
  select(finalWorth,gender) %>%
  filter(!is.na(gender)) %>%
```

```

group_by(gender) %>%
mutate(gender_rank = rank(-finalWorth)) %>%
filter(gender_rank<=100) %>%
ggplot(aes(x=gender_rank,y=finalWorth,color=gender)) +
geom_line()

# d)

forbes %>%
  mutate(young=age<65) %>%
  filter(!is.na(age) & !is.na(gender)) %>%
  mutate(group = paste(if_else(young,"young","old"),
                        if_else(gender=='F','woman','man'), sep="_") ) %>%
  group_by(country,group) %>%
  summarise(n=n()) %>%
  pivot_wider(names_from = group,
              values_from = n,
              values_fill = 0)

# e)
library(lubridate)
forbes %>%
  filter(!(day(birthDate)==1 & month(birthDate)==1)) %>%
  mutate(week=week(birthDate)) %>%
  ggplot(aes(x=week)) +
  geom_bar()

```

II. Predicting high wealth

```

library(tidyverse)
library(tidymodels)

# a)
forbes <- read_csv("forbes_2022_billionaires.csv")
forbes_ml <- forbes %>%
  select(age,finalWorth,age,category,country,selfMade,gender)

forbes_ml %>%
  summarise(across(everything(), is.na)) %>%
  summarise(across(everything(), sum))

forbes_ml <- forbes_ml %>%

```

```
drop_na()
```

```
# b)
```

```
forbes_ml <- forbes_ml %>%  
  mutate(country = country %>% as_factor() %>% fct_lump_min(10))
```

```
# c)
```

```
set.seed(12345)  
forbes_split <- initial_split(forbes_ml, prop=0.75)  
forbes_train <- training(forbes_split)  
forbes_test <- testing(forbes_split)
```

```
# d)
```

```
ols_fit <- linear_reg() %>%  
  set_engine('lm') %>%  
  fit(finalWorth~., data=forbes_train)
```

```
# e)
```

```
xgb_fit <- boost_tree() %>%  
  set_mode("regression") %>%  
  set_engine("xgboost") %>%  
  fit(finalWorth~., data=forbes_train)
```

```
# f)
```

```
pred <- predict(ols_fit, forbes_test) %>%  
  rename(ols_pred = .pred) %>%  
  bind_cols(forbes_test)  
pred <- predict(xgb_fit, forbes_test) %>%  
  rename(xgb_pred = .pred) %>%  
  bind_cols(pred)  
pred %>%  
  mutate(esq_ols=(finalWorth-ols_pred)^2,  
         esq_xgb=(finalWorth-xgb_pred)^2) %>%  
  summarise(mse_ols=mean(esq_ols),  
            mse_xgb=mean(esq_xgb))
```

III. Speed limits and accidents

```
library(tidyverse)
library(numDeriv)

accident <- read_csv('accidents.csv')

L <- function(a,b) {
  x <- accident$speed
  y <- accident$accidents
  ll <- y*(a+b*x)-exp(a+b*x)
  return(sum(ll))
}
L(1,.5)

h <- 1e-6
(L(1+h,.5)-L(1,.5))/h
(L(1,.5+h)-L(1,.5))/h

# Check values
ll <- function(x) {
  return(L(x[1],x[2]))
}
grad(ll,c(1,.5))

#c)

ll <- function(x) {
  return(-L(x[1],x[2]))
}

optim(c(1,0),ll)
```