Practice Exam

Xiang Li(4013115)

2024-01-03

Exercise 1

1

```
cartype_df = read.csv("data/cartype.csv")
head(cartype_df, 5)
```

```
##    Entity Code Year battery_electric_number petrol_number diesel_gas_number
## 1 Austria  AUT 2001                       0        100754            192734
## 2 Austria  AUT 2002                       1         84920            194555
## 3 Austria  AUT 2003                       0         85889            214222
## 4 Austria  AUT 2004                       0         96388            214771
## 5 Austria  AUT 2005                       0        119632            187813
##   hybrid_number
## 1            39
## 2            17
## 3             8
## 4           133
## 5           460
```

The dataset is in long fromat.

2

```
cartype_df1 = cartype_df[cartype_df$Entity %in% c("France", "Germany", "Italy", "Spain",
    "United Kingdom"), ]
head(cartype_df1, 5)
```

```
##    Entity Code Year battery_electric_number petrol_number diesel_gas_number
## 79 France  FRA 2001                     407        986491           1267750
## 80 France  FRA 2002                     233        793425           1351362
## 81 France  FRA 2003                     113        655678           1353419
## 82 France  FRA 2004                     460        619675           1392905
## 83 France  FRA 2005                       6        641022           1423906
##    hybrid_number
## 79            84
```

```
## 80            51
## 81            36
## 82           669
## 83          2855
```

3

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
cartype_df1$tol_number = cartype_df1$battery_electric_number + cartype_df1$petrol_number +
    cartype_df1$diesel_gas_number + cartype_df1$hybrid_number
cartype_df1 = mutate(cartype_df1, battery_electric_prop = battery_electric_number/tol_number,
    petrol_prop = petrol_number/tol_number, diesel_gas_prop = diesel_gas_number/tol_number,
    hybrid_prop = hybrid_number/tol_number)
```

4

```
summarise(group_by(cartype_df1, Year), diesel_pop_coun = Entity[which.max(diesel_gas_prop)])
```

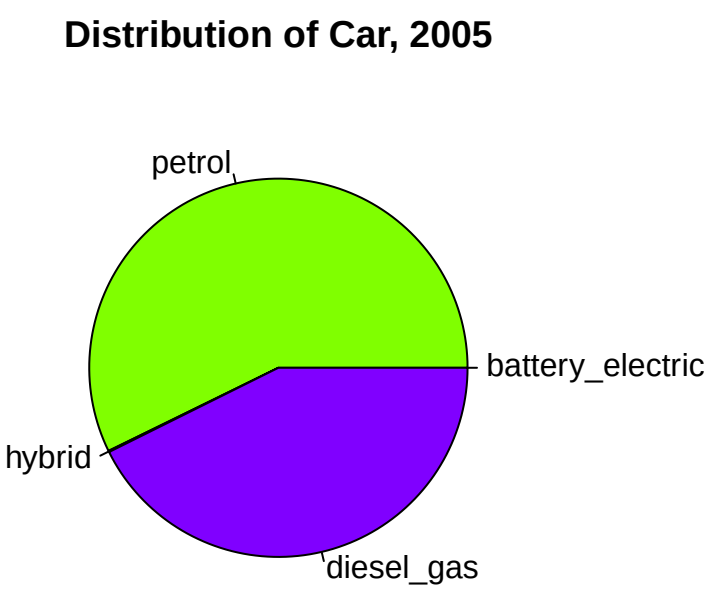
```
## # A tibble: 19 x 2
##     Year diesel_pop_coun
##    <int> <chr>
##  1  2001 France
##  2  2002 France
##  3  2003 France
##  4  2004 France
##  5  2005 Spain
##  6  2006 France
##  7  2007 France
##  8  2008 France
##  9  2009 France
## 10  2010 Spain
## 11  2011 France
## 12  2012 France
## 13  2013 Spain
## 14  2014 Spain
## 15  2015 Spain
```

```
## 16  2016 Italy
## 17  2017 Italy
## 18  2018 Italy
## 19  2019 Italy
```

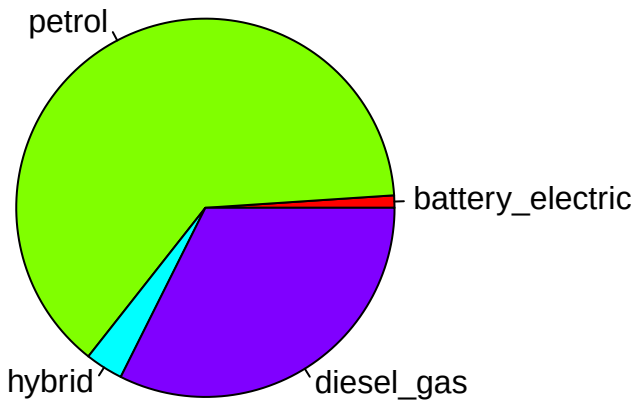
Based on the table above, diesel cars were most popular in France in 2008 and in Italy in 2018.

5

```
G2005_df = data.frame(type = c("battery_electric", "petrol", "hybrid", "diesel_gas"),
    number = unlist(cartype_df1[(cartype_df1$Entity == "Germany") & (cartype_df1$Year ==
        2005), c("battery_electric_number", "petrol_number", "hybrid_number", "diesel_gas_number")]))
pie(x = G2005_df$number, labels = G2005_df$type, col = rainbow(4), main = "Distribution of Car, 2005")
```




```
G2018_df = data.frame(type = c("battery_electric", "petrol", "hybrid", "diesel_gas"),
    number = unlist(cartype_df1[(cartype_df1$Entity == "Germany") & (cartype_df1$Year ==
        2018), c("battery_electric_number", "petrol_number", "hybrid_number", "diesel_gas_number")]))
pie(x = G2018_df$number, labels = G2018_df$type, col = rainbow(4), main = "Distribution of Car, 2018")
```

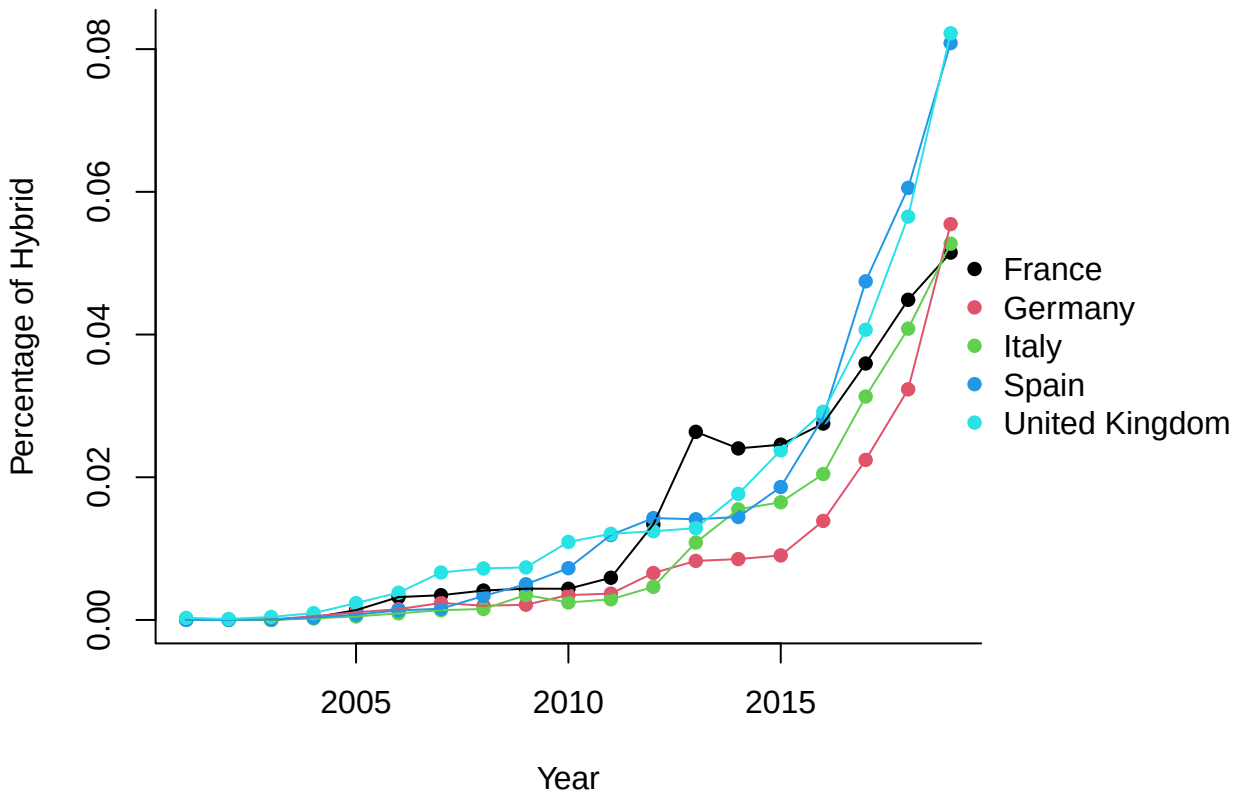


The proportions of fully electric battery vehicles and hybrid vehicles increase from 2005 to 2018.

6

```
library(ptmixed)
```

```
make.spaghetti(Year, hybrid_prop, id = Entity, group = Entity, data = cartype_df1,
    col = 1:5, ylab = "Percentage of Hybrid", legend.inset = -0.32)
```

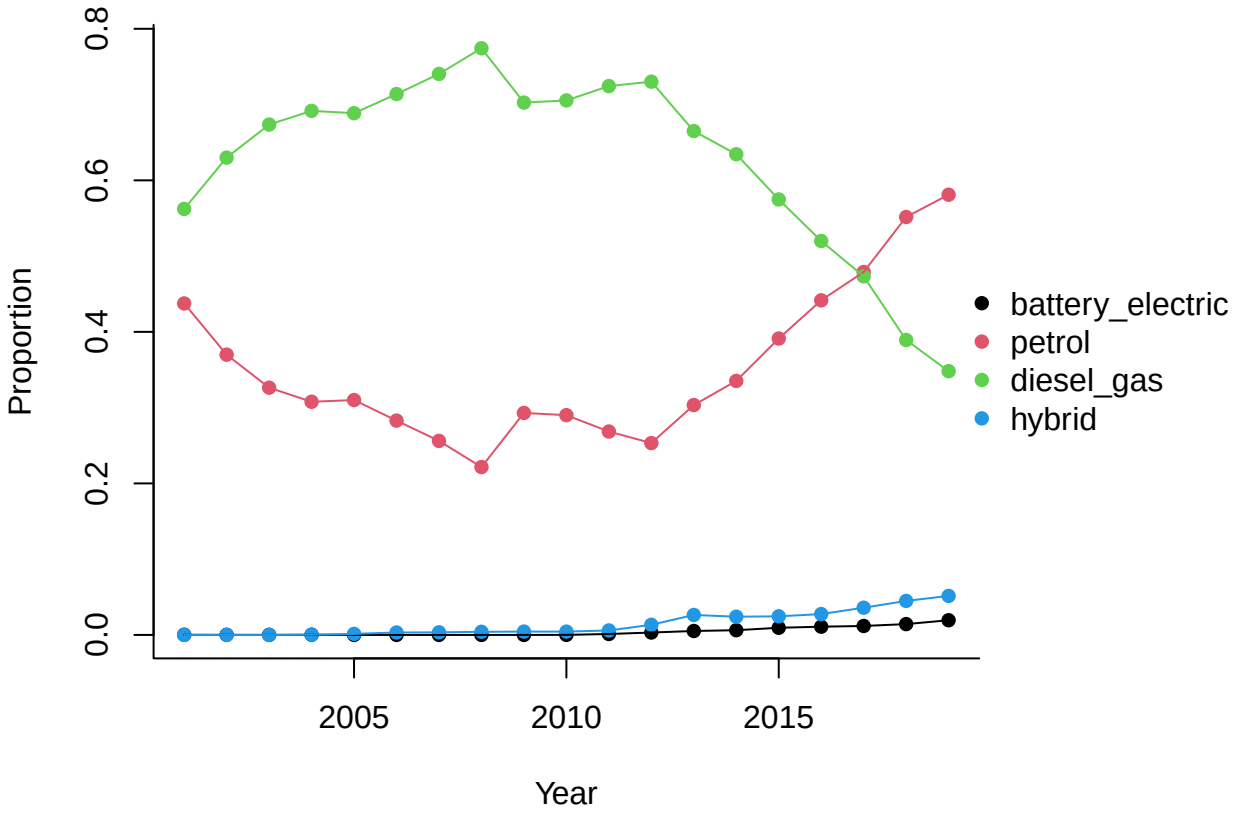


The line of United Kingdom is most steep. The United Kingdom has seen a faster diffusion of hybrid vehicles.

7

```
library(reshape2)
```

```
Fran_df = cartype_df1[cartype_df1$Entity == "France", c(3, 9:12)]
Fran_df = rename(Fran_df, battery_electric = battery_electric_prop, petrol = petrol_prop,
    diesel_gas = diesel_gas_prop, hybrid = hybrid_prop)
Fran_long_df = melt(Fran_df, id.vars = "Year")
Fran_long_df = rename(Fran_long_df, Type = variable, Proportion = value)
make.spaghetti(Year, Proportion, id = Type, group = Type, data = Fran_long_df, col = 1:4,
    legend.inset = -0.32)
```

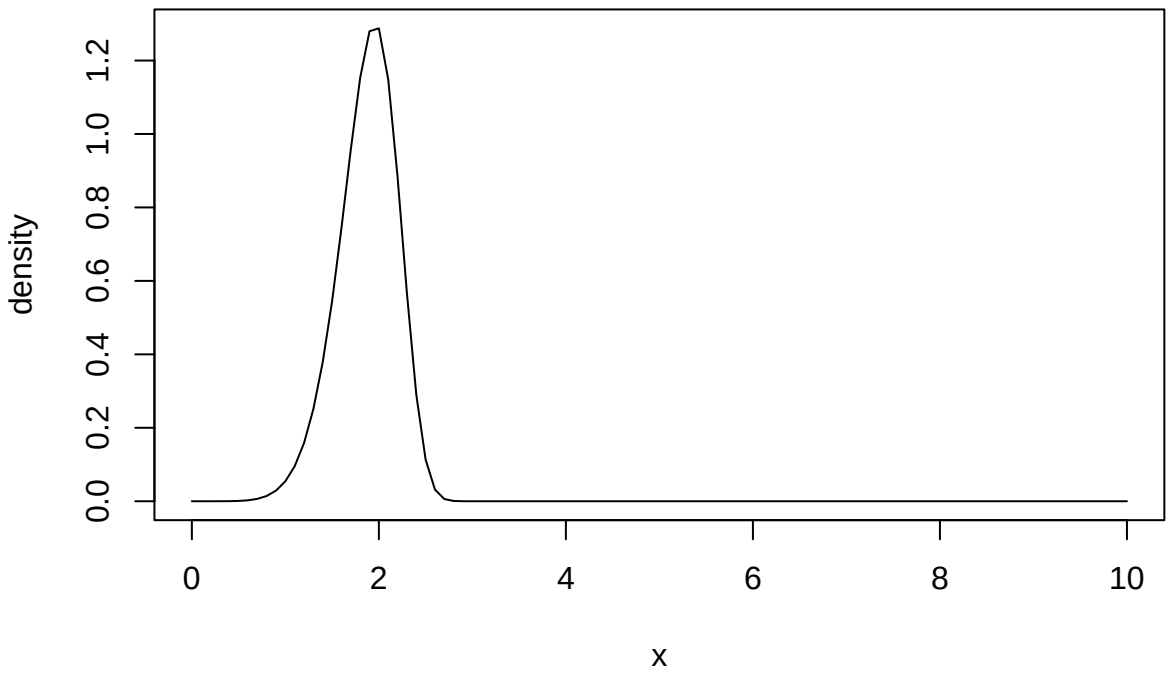


From 2005 to 2019 in France, the proportions of fully electric battery vehicles and hybrid vehicles slowly increase. The proportion of diesel vehicles first increase then decrease, and the proportion of petroleum vehicles first decrease then increase. I guess diesel vehicles were replaced by petroleum vehicles.

Exercise 2

1

```
df = function(x, k, lamb) {
    (k/lamb) * ((x/lamb)^(k - 1)) * exp(-((x/lamb)^k))
}
curve(df(x, k = 7, lamb = 2), from = 0, to = 10, ylab = "density")
```



2

```
optimize(df, interval = c(0.01, 30), k = 3, lamb = 4, maximum = TRUE)
```

```
## $maximum
## [1] 3.494313
##
## $objective
## [1] 0.2938579
```

The mode for $k = 3$ and $\lambda = 4$ is around 3.494.

3

```
optimize(df, interval = c(0.01, 30), k = 2.3, lamb = 5.7, maximum = TRUE)
```

```
## $maximum
## [1] 4.447773
##
## $objective
## [1] 0.1660849
```

The mode for $k = 2.3$ and $\lambda = 5.7$ is around 4.448.

4

The answer is by pen and paper.

5

```
mode_f = function(k, lamb) {
    lamb * (((k - 1)/k)^(1/k))
}
mode_f(k = 3, lamb = 4)
```

```
## [1] 3.494322
```

```
mode_f(k = 2.3, lamb = 5.7)
```

```
## [1] 4.447772
```

The analytical results are same with numerical results.

Advantage: easy and save time.

Disadvantage: it sometimes can't find the accurate solution.

Exercise 3

1

```
getSummaries = function(x) {
    result = data.frame(Statistic = c("Sample size", "Mean", "Median", "Variance"),
        Value = c(length(x), mean(x), median(x), var(x)))
    return(result)
}
```

2

```
getSummaries = function(x, digits) {
    result = data.frame(Statistic = c("Sample size", "Mean", "Median", "Variance"),
        Value = c(length(x), mean(x), median(x), var(x)))
    result$Value = round(result$Value, digits)
    return(result)
}
```

3

```r
getSummaries = function(x, digits, ignoreNAs) {
    if (ignoreNAs) {
        result = data.frame(Statistic = c("Sample size", "Sample size without NA",
            "Mean", "Median", "Variance"), Value = c(length(x), length(x[!is.na(x)]),
            mean(x, na.rm = T), median(x, na.rm = T), var(x, na.rm = T)))
    } else {
        warning("There could be NAs in x!!!\n")
        result = data.frame(Statistic = c("Sample size"), Value = c(length(x)))
    }
    result$Value = round(result$Value, digits)
    return(result)
}
```

4

```r
getSummaries_v = function(x, digits, ignoreNAs) {
    if (ignoreNAs) {
        result = data.frame(Statistic = c("Sample size", "Sample size without NA",
            "Mean", "Median", "Variance"), Value = c(length(x), length(x[!is.na(x)]),
            mean(x, na.rm = T), median(x, na.rm = T), var(x, na.rm = T)))
    } else {
        warning("There could be NAs in x!!!\n")
        result = data.frame(Statistic = c("Sample size"), Value = c(length(x)))
    }
    result$Value = round(result$Value, digits)
    return(result)
}

getSummaries = function(x, digits, ignoreNAs) {
    if (is.matrix(x)) {
        result = apply(x, 2, getSummaries_v, digits = digits, ignoreNAs = ignoreNAs,
            simplify = F)
        return(result)
    } else if (is.vector(x)) {
        return(getSummaries_v(x, digits = digits, ignoreNAs = ignoreNAs))
    }
}
```

5

```r
getSummaries = function(x, digits, ignoreNAs) {
    if (is.matrix(x)) {
        result = apply(x, 2, getSummaries_v, digits = digits, ignoreNAs = ignoreNAs,
            simplify = F)
        return(result)
    } else if (is.vector(x)) {
        return(getSummaries_v(x, digits = digits, ignoreNAs = ignoreNAs))
    } else {
        stop("x is invalid!!!\n")
```

```
    }
}
```

```
set.seed(3078)
x1 = rhyper(300, 5, 20, 10)
x2 = c(rep(cars$speed, 3), rep(NA, 50), rep(cars$dist, 2))
x3 = cbind(x1, x2)
```

6

```
getSummaries(x1, digits = 2, ignoreNAs = T)
```

```
##                Statistic  Value
## 1            Sample size 300.00
## 2 Sample size without NA 300.00
## 3                   Mean   1.96
## 4                 Median   2.00
## 5               Variance   1.01
```

```
getSummaries(x2, digits = 2, ignoreNAs = T)
```

```
##                Statistic  Value
## 1            Sample size 300.00
## 2 Sample size without NA 250.00
## 3                   Mean  26.43
## 4                 Median  19.00
## 5               Variance 461.15
```

```
getSummaries(x3, digits = 2, ignoreNAs = T)
```

```
## $x1
##                Statistic  Value
## 1            Sample size 300.00
## 2 Sample size without NA 300.00
## 3                   Mean   1.96
## 4                 Median   2.00
## 5               Variance   1.01
##
## $x2
##                Statistic  Value
## 1            Sample size 300.00
## 2 Sample size without NA 250.00
## 3                   Mean  26.43
## 4                 Median  19.00
## 5               Variance 461.15
```

7

```r
getSummaries(x1, digits = 4, ignoreNAs = F)
```

```
## Warning in getSummaries_v(x, digits = digits, ignoreNAs = ignoreNAs): There could be NAs in x!!!
```

```
##     Statistic Value
## 1 Sample size   300
```

```r
getSummaries(x2, digits = 4, ignoreNAs = F)
```

```
## Warning in getSummaries_v(x, digits = digits, ignoreNAs = ignoreNAs): There could be NAs in x!!!
```

```
##     Statistic Value
## 1 Sample size   300
```

```r
getSummaries(x3, digits = 4, ignoreNAs = F)
```

```
## Warning in FUN(newX[, i], ...): There could be NAs in x!!!

## Warning in FUN(newX[, i], ...): There could be NAs in x!!!
```

```
## $x1
##     Statistic Value
## 1 Sample size   300
##
## $x2
##     Statistic Value
## 1 Sample size   300
```