name = 황창현

student id = 17010668

1. table

1-1. table 1

```
# table 1

products=read.csv("./csv/products(1).csv",stringsAsFactors=F,na.strings=c("NA","NULL"))
od=read.csv("./csv/orderdetails.csv",stringsAsFactors=F,na.strings=c("NA","NULL"))

merge1=products %>% inner_join(od) %>%
group_by(productCode) %>%
summarise(SumOfQuantityOrdered=sum(quantityOrdered),
Average_Sales_Price=mean(priceEach)) %>%
arrange(desc(SumOfQuantityOrdered))

x=c("productName","productCode","productVendor","SumOfQuantityOrdered")
merge2=select(products,productCode,productName,productVendor,productLine)

data2=merge1 %>% inner_join(merge2)
data=slice(data2[x],1:10)
```

```
$A tibble: 10 x 4
$ productName                          productCode productVendor          SumOfQuantityOrder~
$ <chr>                                <chr>       <chr>                                <int>
$ 1992 Ferrari 360 Spider red          S18_3232    Unimax Art Galleries                  1808
$ 1937 Lincoln Berline                 S18_1342    Motor City Art Classi~                1111
$ American Airlines: MD-11S            S700_4002   Second Gear Diecast                   1085
$ 1941 Chevrolet Special Deluxe Cabrio~ S18_3856   Exoto Designs                         1076
$ 1930 Buick Marquette Phaeton         S50_1341    Studio M Art Models                   1074
$ 1940s Ford truck                     S18_4600    Motor City Art Classi~                1061
$ 1969 Harley Davidson Ultimate Chopper S10_1678   Min Lin Diecast                       1057
$ 1957 Chevy Pickup                    S12_4473    Exoto Designs                         1056
```

설명 : 데이터를 불러온 후, 한번에 하는 방법을 몰라서 merge1과 merge2로 나눴다.

merge1 은 products 와 orderdetails를 inner_join한 후 필요한 column을 계산하고 내림차순 까지만 했다. 뒤에 그래프 그릴 때 all product를 그려야 하기 때문에 나뒀다.

x는 배열로 dataframe 열 순서를 그림과 같이 맞추기위해서 따로 지정했고,

merge2 는 select 로 productline 그래프 그릴 때 필요한 column 까지 가져왔다.

그리고 merge2에서 merge1에 필요한 내용만 inner_join 한후 슬라이스 했다.

1-2. table 2

```
# table 2

customers=read.csv("./csv/customers.csv",stringsAsFactors=F,na.strings=c('NA','NULL'))
od=read.csv("./csv/orderdetails.csv",stringsAsFactors=F,na.strings=c('NA','NULL'))
orders=read.csv("./csv/orders.csv",stringsAsFactors=F,na.strings=c('NA','NULL'))

merge1=merge(customers,orders,by='customerNumber')
merge2=merge(merge1,od,by='orderNumber') %>%
select(customerNumber,customerName,city,country,quantityOrdered,priceEach,creditLimit)

merge3=merge2 %>% group_by(customerNumber) %>%
summarise(Amount=sum(quantityOrdered*priceEach))

y=c("customerNumber","customerName","city","country","creditLimit")
merge2=distinct(merge2[y])
merge4=merge3 %>% inner_join(merge2)
x=c("customerNumber","customerName","city","country","Amount")
data=merge4[x] %>%
arrange(desc(Amount)) %>% slice(1:10)
```

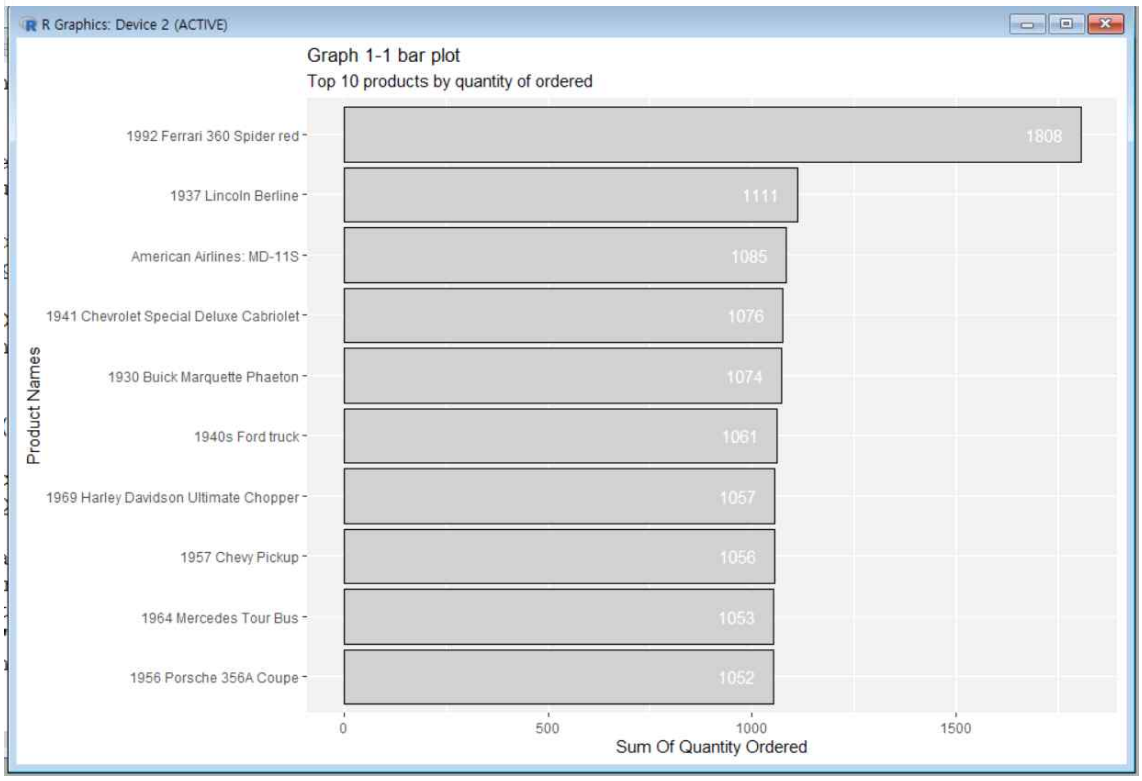
```
# A tibble: 10 x 5
   customerNumber customerName                city          country      Amount
            <int> <chr>                       <chr>         <chr>         <dbl>
 1            141 Euro+ Shopping Channel      "Madrid"      Spain        820690.
 2            124 Mini Gifts Distributors Ltd. "San Rafael" USA          591827.
 3            114 Australian Collectors, Co.  "Melbourne"   Australia    180585.
 4            151 Muscle Machine Inc          "NYC"         USA          177914.
 5            119 La Rochelle Gifts           "Nantes"      France       158573.
 6            148 Dragon Souveniers, Ltd.     "Singapore"   Singapore    156251.
 7            323 Down Under Souveniers, Inc  "Auckland  "  New Zealand  154622.
 8            131 Land of Toys Inc.           "NYC"         USA          149085.
 9            187 AV Stores, Co.              "Manchester"  UK           148410.
10            450 The Sharp Gifts Warehouse   "San Jose"    USA          143536.
> 
```

필요한 데이터를 읽고 sql 구문에 따라서 inner_join을 하려 했는데 너무 어려워서 구글링해서 merge 했다. merge로 한쪽에 inner_join 이 아닌, 전체에서 한번에 자르기 위해서 이다.
merge2 는 merge 한 후 바로 select로 필요한 부분만 꺼냈다.
merge3 는 Amount를 꺼내는 중간 dataframe 이고, merge2가 merge3의 부모격이니 merge2의 priceEach 와 quantityOrdered를 제외한 부분을 distinct으로 뽑아냈다. 그리고 다시 merge3 와 merge2가 inner_join 한 것이 merge4다.
merge4 까지는 전체 데이터가 있으니 data 변수에 merge4에서 내림차순으로 하고 top 10 만 자르는 전처리를 해서 할당했다.

2. graph

1-1

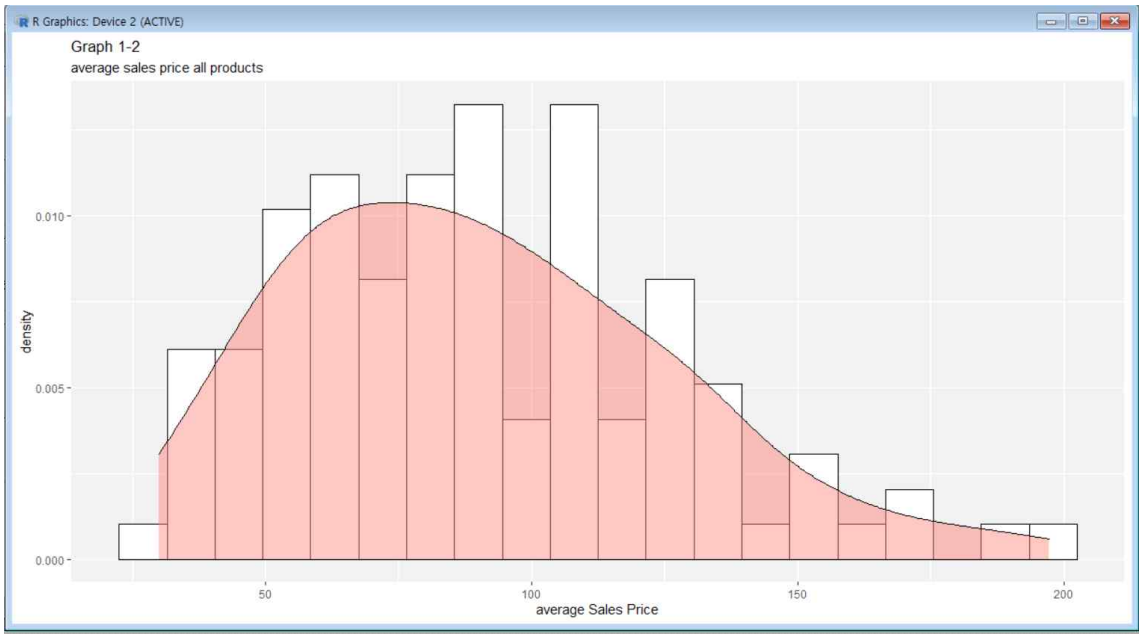
```
# 1
ggplot(data,aes(y=reorder(productName,SumOfQuantityOrdered),
x=SumOfQuantityOrdered))+
geom_bar(stat="identity",colour='black',fill='#CCCCCC')+
geom_text(aes(label=SumOfQuantityOrdered),
color='white',hjust=1.5)+
labs(title = "Graph 1-1 bar plot",
       subtitle="Top 10 products by quantity of ordered",
       x = "Average Sales Price",
       y = "Product Names",
       color = "productLine")
```



bar plot 은 data 로도 할 수 있어서 data로 했다.
aes에 x 수직 막대 대신 y 수평막대를 넣었고, 정렬 하는 방법을 몰라 구글링 해서 reorder 했다.
색깔 맞추기 위해서 edge color = colour 로 블랙 맞추고 fill = color 로 회색 넣었다.
그리고 글씨 넣는 법도 몰라서 구글링했다.
geom_text 로 hjust > 0 로 왼쪽으로 땡겼다. < 0 면 오른쪽으로 땡겨진다.
그 외 글씨들은 최대한 가깝게 했다.

1-2

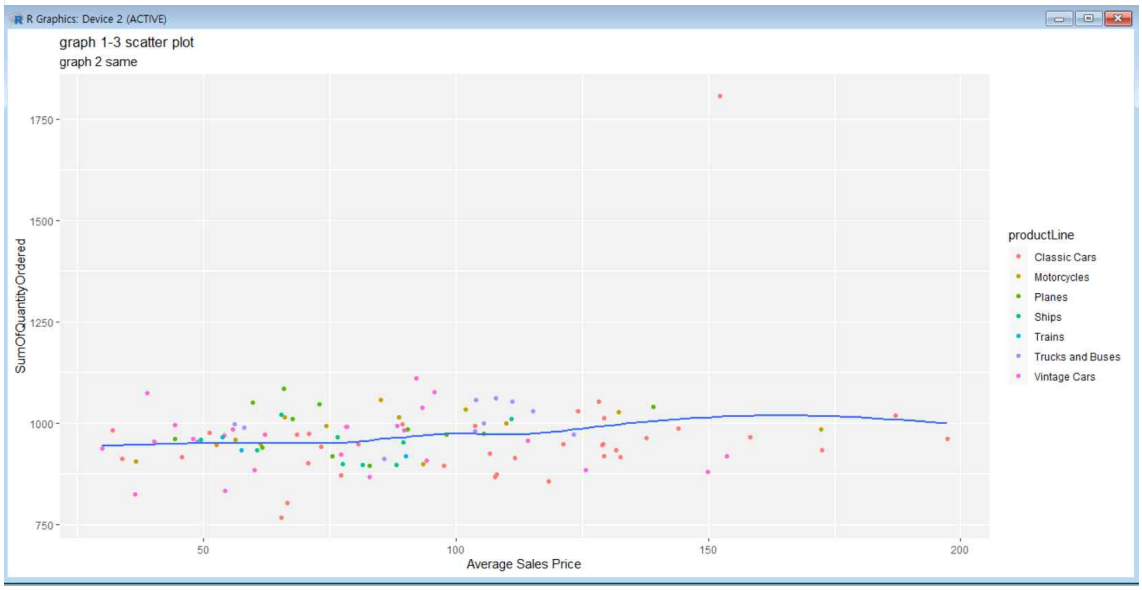
```
# 2
ggplot(data2,aes(x=Average_Sales_Price))+
geom_histogram(aes(y=..density..),fill='white',
colour='black',binwidth=9)+
geom_density(fill='#f8867d',alpha=0.5)+
labs(title="Graph 1-2",
subtitle="average sales price all products",
x="average Sales Price",
y="density")
```



2번 plot부터 data2 데이터를 사용했다. average Sales Price 같은 경우는 summarise로 mean(priceEach)를 통해 얻었고, table 1 에 보여지지 않는 부분이기 때문에 data2로 놓았다. 위 그래프는 수업 scripts에 없어서 구글링해서 찾았다. barwidth를 그림과 맞추기 위해서 숫자를 임의 조정했고, 색깔 글자 등 맞췄다.

1-3

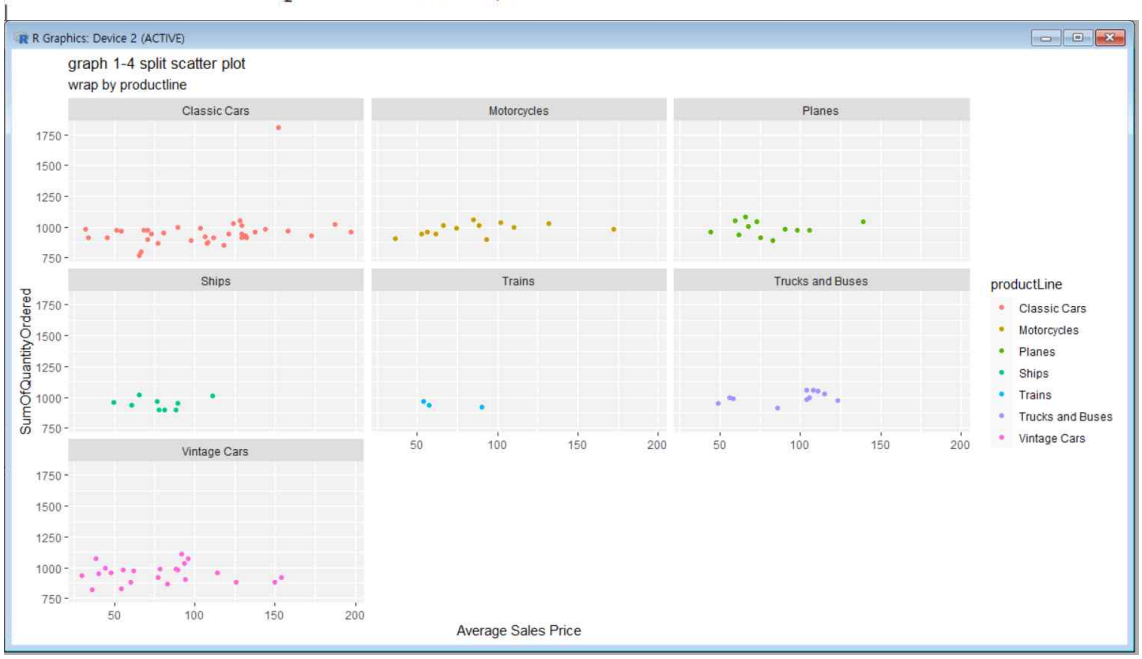
```
# 3
ggplot(data2, aes(x = Average_Sales_Price, y = SumOfQuantityOrdered)) +
geom_point(aes(color = productLine)) +
geom_smooth(se = FALSE)+
labs(title = "graph 1-3 scatter plot",
       subtitle = "graph 2 same",
       x = "Average Sales Price",
       y = "SumOfQuantityOrdered",
       color = "productLine")
```



위 3번 plot도 data2를 사용했고, scatter plot이라 geom_point를 사용했다.
후에, 신뢰구간선도 구해야 해서 smooth를 썼고, 95% 는 필요없어서 se=FALSE로 했다.
그 외 그림과 같이 최대한 맞췄다.

1-4

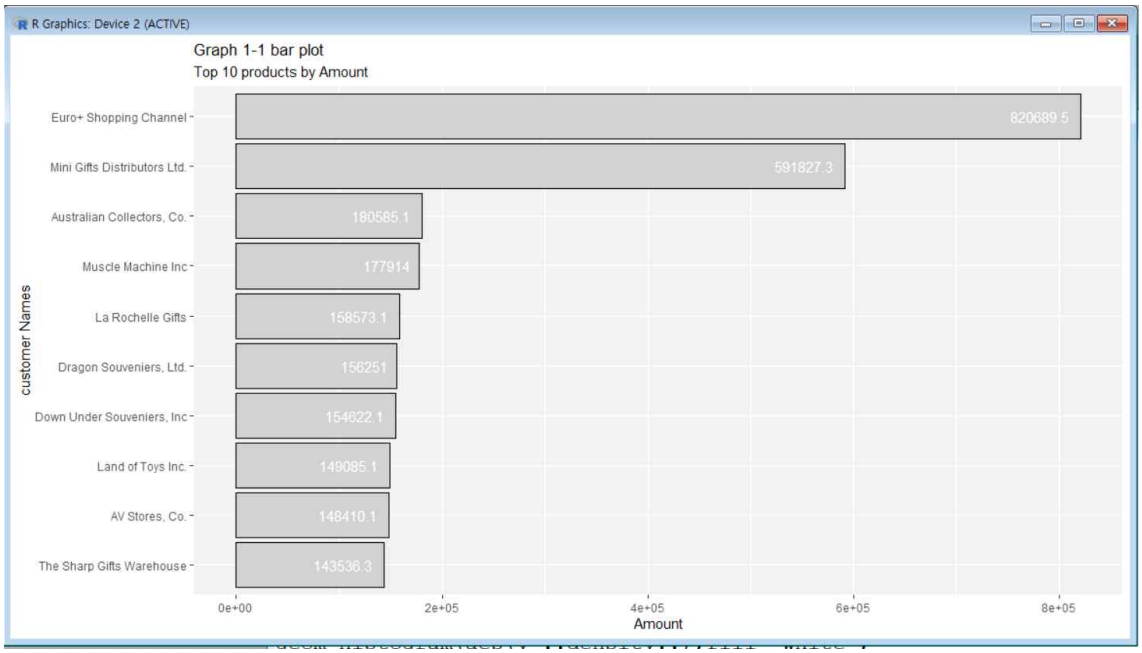
```
# 4
ggplot(data2,aes(x=Average_Sales_Price,y=SumOfQuantityOrdered))+
geom_point(aes(color=productLine))+
facet_wrap(~productLine)+
labs(title = "graph 1-4 split scatter plot",
       subtitle="wrap by productline",
       x = "Average Sales Price",
       y = "SumOfQuantityOrdered",
       color = "productLine")
```



faced 그래프에는 grid 와 wrap 이 있다. grid 는 세로로 나눠가지고 시각화에 굉장히 안좋다. 그래서 wrap 으로 했다. 그 외는 문법은 그림과 최대한 가깝게 맞추려고 했다.

2-1

```
data$Amount=round(data$Amount,1)
ggplot(data,aes(y=reorder(customerName,Amount),
x=Amount))+
geom_bar(stat="identity",colour='black',fill='#CCCCCC')+
geom_text(aes(label=Amount),
color='white',hjust=1.2)+
labs(title = "Graph 1-1 bar plot",
       subtitle="Top 10 products by Amount",
       x = "Amount",
       y = "customer Names")
```

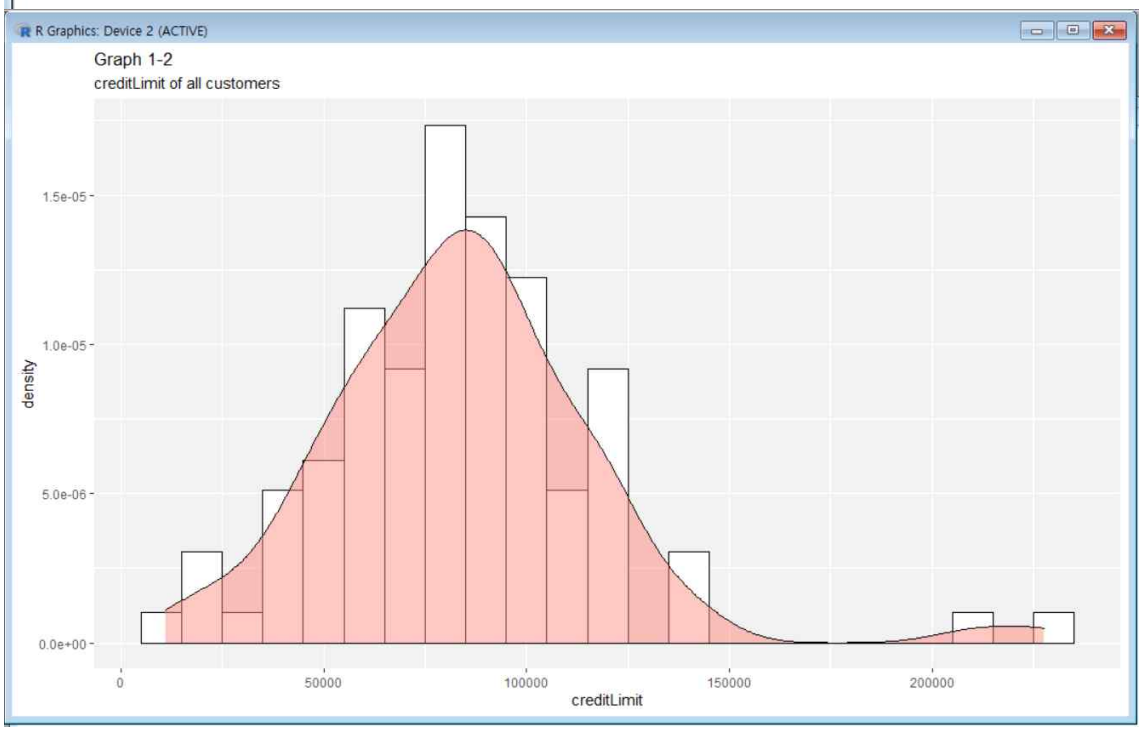


일단, 전체적인 코드 양상은 첫 번째와 같다.

한가지는 숫자가 소수점 첫째자리라서 round 함수를 적용했다.

2-2

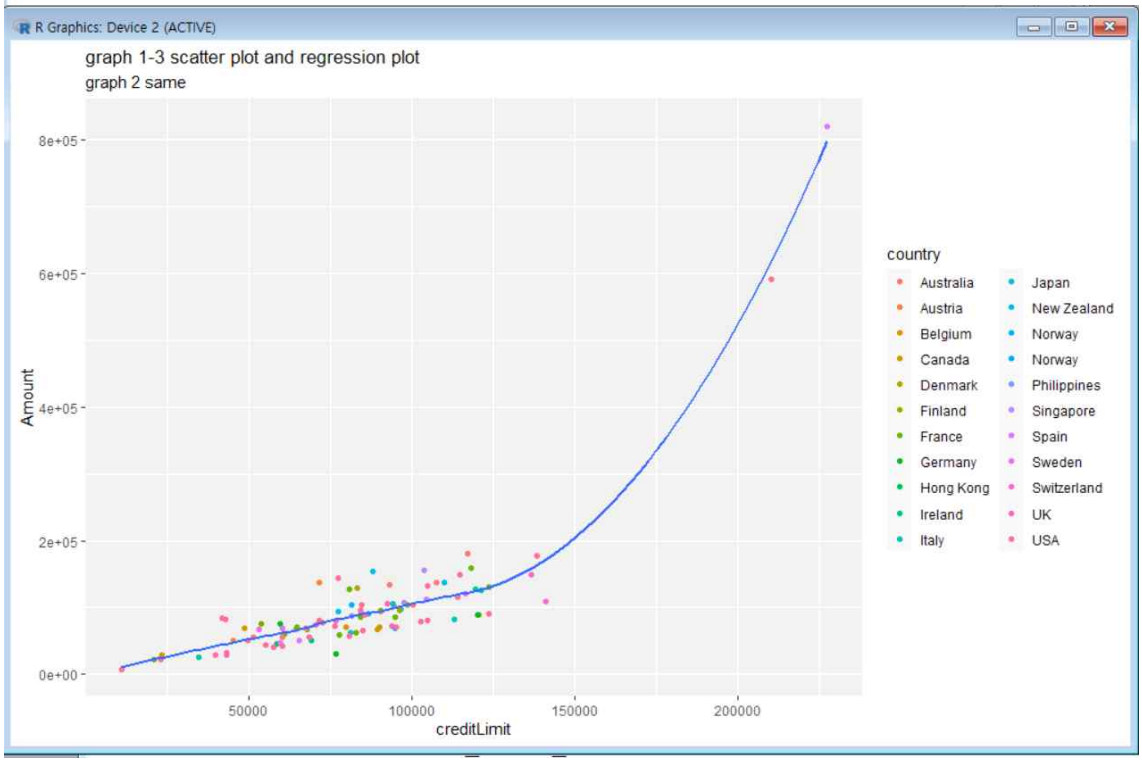
```
# 2
ggplot(merge4,aes(x=creditLimit))+
geom_histogram(aes(y=..density..),fill='white',
colour='black',binwidth=10000)+
geom_density(fill='#f8867d',alpha=0.5)+
labs(title="Graph 1-2",
subtitle="creditLimit of all customers",
x="creditLimit",
y="density")
```



위와 같다. 이번엔 밑의 x 범위가 넓기 때문에 barwidth를 10000으로 했고, 그림과 얼추 맞았다.

2-3.

```
ggplot(merge4, aes(x = creditLimit, y = Amount)) +
geom_point(aes(color = country)) +
geom_smooth(se = FALSE)+
labs(title = "graph 1-3 scatter plot and regression plot",
       subtitle = "graph 2 same",
       x = "creditLimit",
       y = "Amount",
       color = "country")
```



위와 같다.

2-4

```
# 4
ggplot(merge4,aes(x=creditLimit,y=Amount))+
geom_point(aes(color=country))+
facet_wrap(~ country)+
labs(title = "graph 1-4 split scatter plot",
       subtitle="wrap by country",
       x = "credit Limit",
       y = "Amount",
       color = "country")
```

위와 같다.

3. discussion

일단 교수님이 해주신 것처럼 똑같이 해보려고 했지만, 시간이 오래걸리고 어렵다는 것을 알았다. 교수님이 해주신건 눈에 잘 들어오지만, 내껀 아니다.

색이 중요함을 알았고, 잘 보이기 위해서 가끔은 글씨를 써야 할 때도 그리고 지워야 할 때도 있음을 알았다.

얼핏보면 관련없어 보이는 변수들이 그래프로 그렸을 때 서로 연관이 있을 수 있다는 것을 알았고, 하나하나 뜯어보면서 분석해야함을 알았다.

HW2부터 HW3 까지 SQL부터 R 까지 데이터 추출부터 분석까지 기초적으로 할 수 있어 좋았다.