Analysis of Farmers Market in USA

```
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
```

```
library(usmap)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────────────────────────
──────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.2.1     ✓ purrr   0.3.3
## ✓ tibble  2.1.3     ✓ stringr 1.4.0
## ✓ tidyr   1.0.2     ✓ forcats 0.4.0
## ✓ readr   1.3.1
```

```
## ── Conflicts ────────────────────────────────────────────────────────────────
──────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

Farmers Markert Directory

The Farmers Market Directory lists markets selling agricultural products directly to customers at a common, recurrent physical location. Maintained by the Agricultural Marketing Service, the Directory is designed to provide customers with convenient access to information about farmers market listings to include: market locations, operating times, product offerings, accepted forms of payment, and more

```
fmarket <- read.csv("fmarket.csv",header = T, stringsAsFactors = FALSE)
```

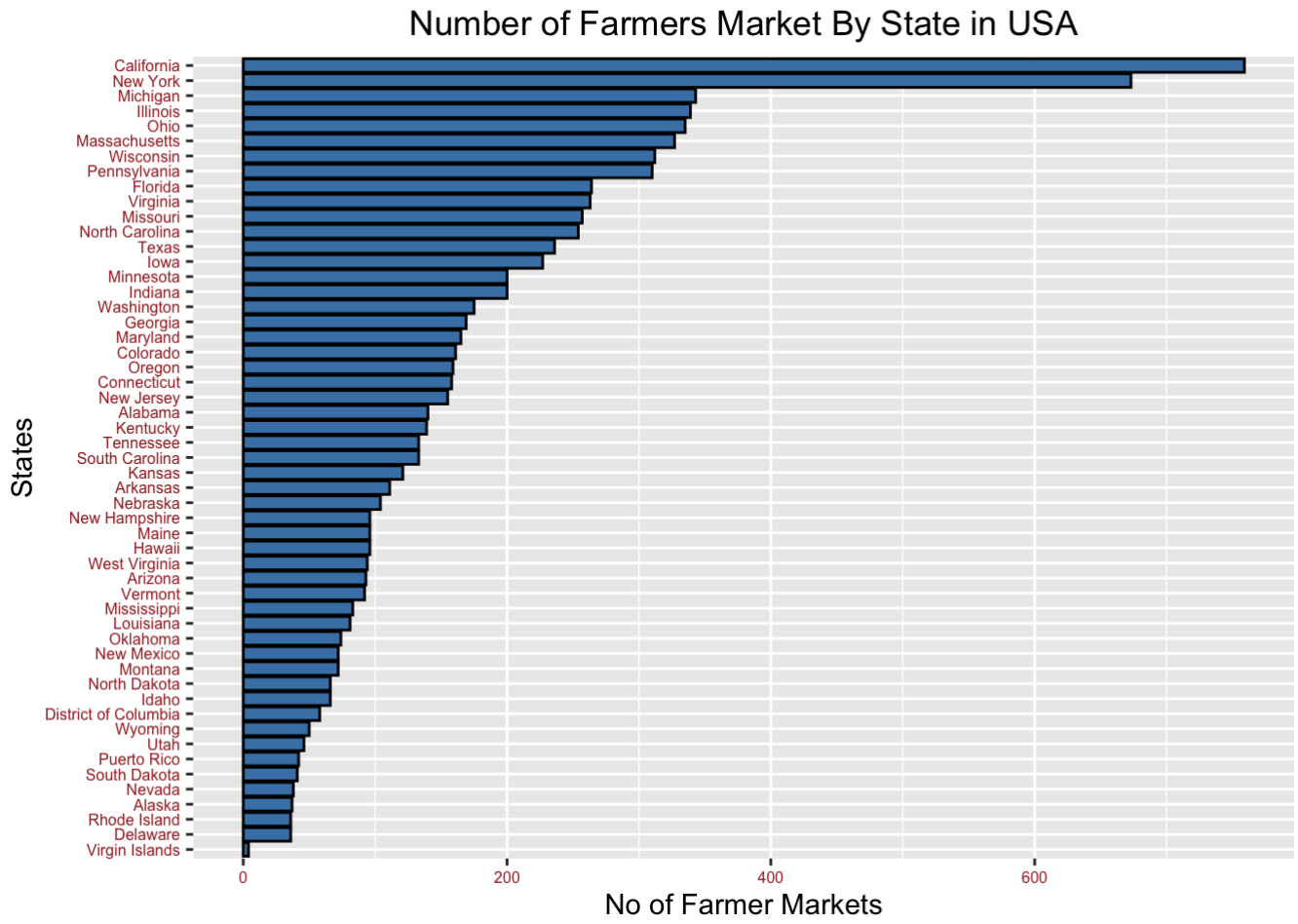
Farmers ' markets are generally considered to be seasonal markets where farm products are sold by farmers themselves at fixed locations. Although other locations, such as' public' or' municipal markets," terminal markets," farm shops," farm stands," curb' or' tailgate markets," flea markets,' and' swap meets,' can sometimes be called' farmers ' markets,' some, if not all, of the vendors must be producers selling their own products.

Count of Markets by State in USA

State-wise count of the markets gives you a clear idea of the total available markets and also provides you with information on the future opportunities for new markets.

```
#Counting the total number of farmers markets  by state in USA

CountState <-aggregate(list(Count = fmarket$FMID),by = list(states = fmarket$State), FUN
= length)

ggplot(data = CountState,aes(x = reorder(states,Count),y = Count)) +
  geom_bar(stat="identity",color = "black", fill = "Steelblue") +
  coord_flip() +
  labs(x="States", y="No of Farmer Markets") +
  ggtitle("Number of Farmers Market By State in USA") +
  theme(axis.text = element_text(color="brown", size=6),plot.title = element_text(hjust
 = 0.5)) +
  ggsave("FmarketVsState.png", device="png", dpi = 600)
```

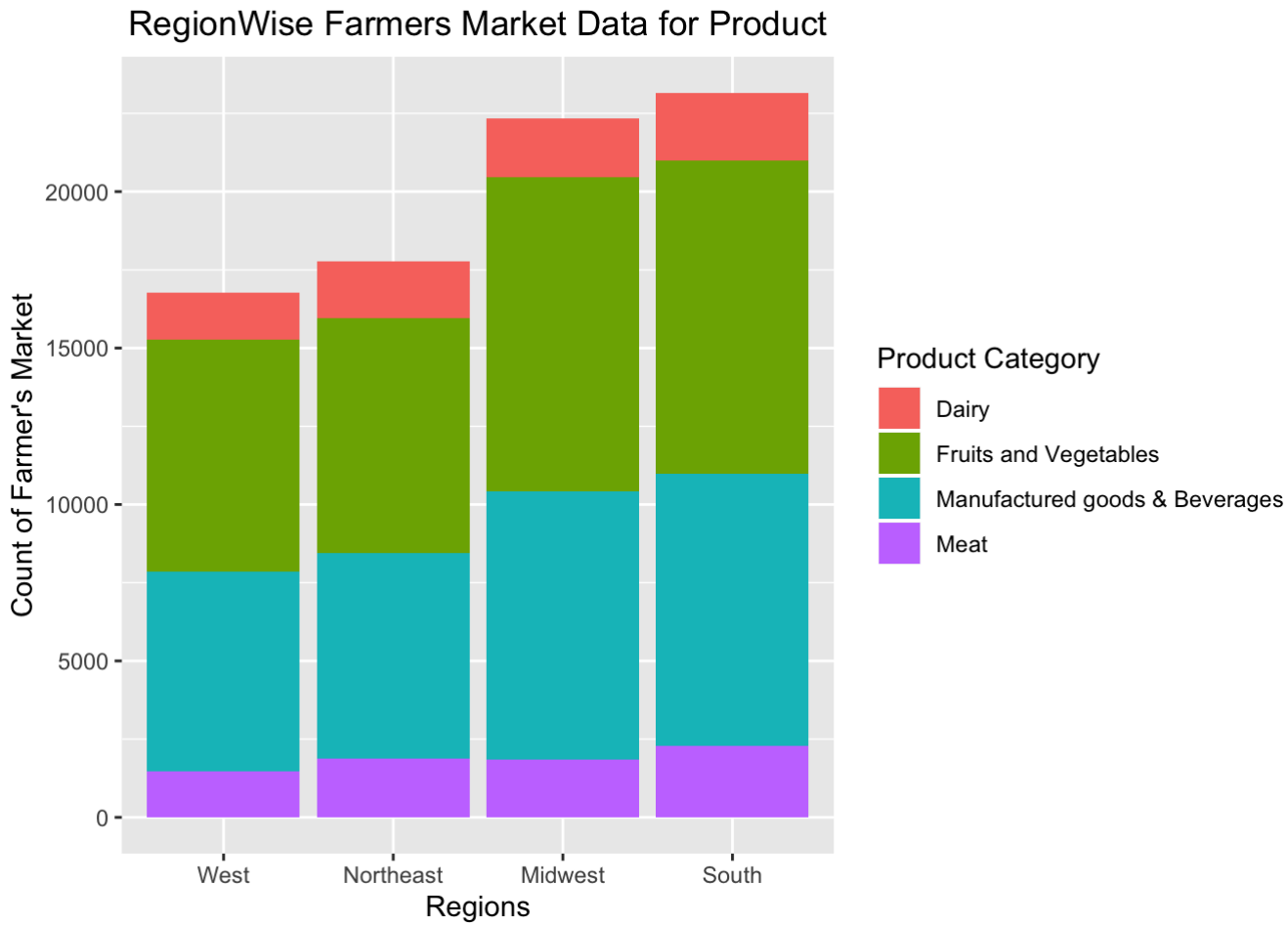
```
## Saving 7 x 5 in image
```



Statcked bar graph for region and product

```
#Modification in Data.

#Importing new file to map region and state data. There are 52 states and plotting them
 is difficult. Hence considering regions in plotting some plots. Also number of food pro
ducts are many so clubbing them to create a supertype for them

StatesInfo <-read.csv("States.csv",as.is = TRUE)
Fmarket_States <-merge(y=StatesInfo,x=fmarket, by.x = "State", by.y = "State",all.x = TR
UE)

product<-melt(Fmarket_States[,c(1,2,30:58,61)], id.vars = c("FMID","Region"))
product<-product[product$value=='Y',]

#Created custom caregories for all the products as number of products are too many
FruitsVegetables<-c("Vegetables","Plants","Beans","WildHarvested","Fruits","Grains","Flo
wers","Maple","Trees","Mushrooms","Nursery","Herbs","Nuts","Coffee")
Meat <-c("Seafood","Meat","Poultry")
Dairy<-c("Cheese","Eggs")
AllKindBeverages<-c("Bakedgoods","Honey","Jams","Juices","Soap","PetFood","Tofu","Prepar
ed","Crafts","Wine")


colnames(product)[3] <- "OldCategory"

product$NewCategory<-NA
product[product$OldCategory %in% FruitsVegetables,5]="Fruits and Vegetables"
product[product$OldCategory %in% Meat,5]="Meat"
product[product$OldCategory %in% Dairy,5]="Dairy"
product[product$OldCategory %in% AllKindBeverages,5]="Manufactured goods & Beverages"

#Summarizing by NewCategory and Region
CategoryCount <- product %>%
  group_by(NewCategory,Region)%>%
  dplyr::summarise(Ccount=n()) %>%
  arrange(desc(Ccount))

#Omiting all NA's
CategoryCount <- na.omit(CategoryCount)

ggplot(CategoryCount, aes(fill=NewCategory, y=Ccount, x=reorder(Region,Ccount))) +
  geom_bar(position="stack", stat="identity") +
  labs(x = "Regions", y= "Count of Farmer's Market", title="RegionWise Farmers Market Da
ta for Product", fill = "Product Category") +
  theme(legend.position = "right", plot.title = element_text(hjust = 0.5)) +
  ggsave("ProductVsFmarket.png", device="png", dpi = 600)
```

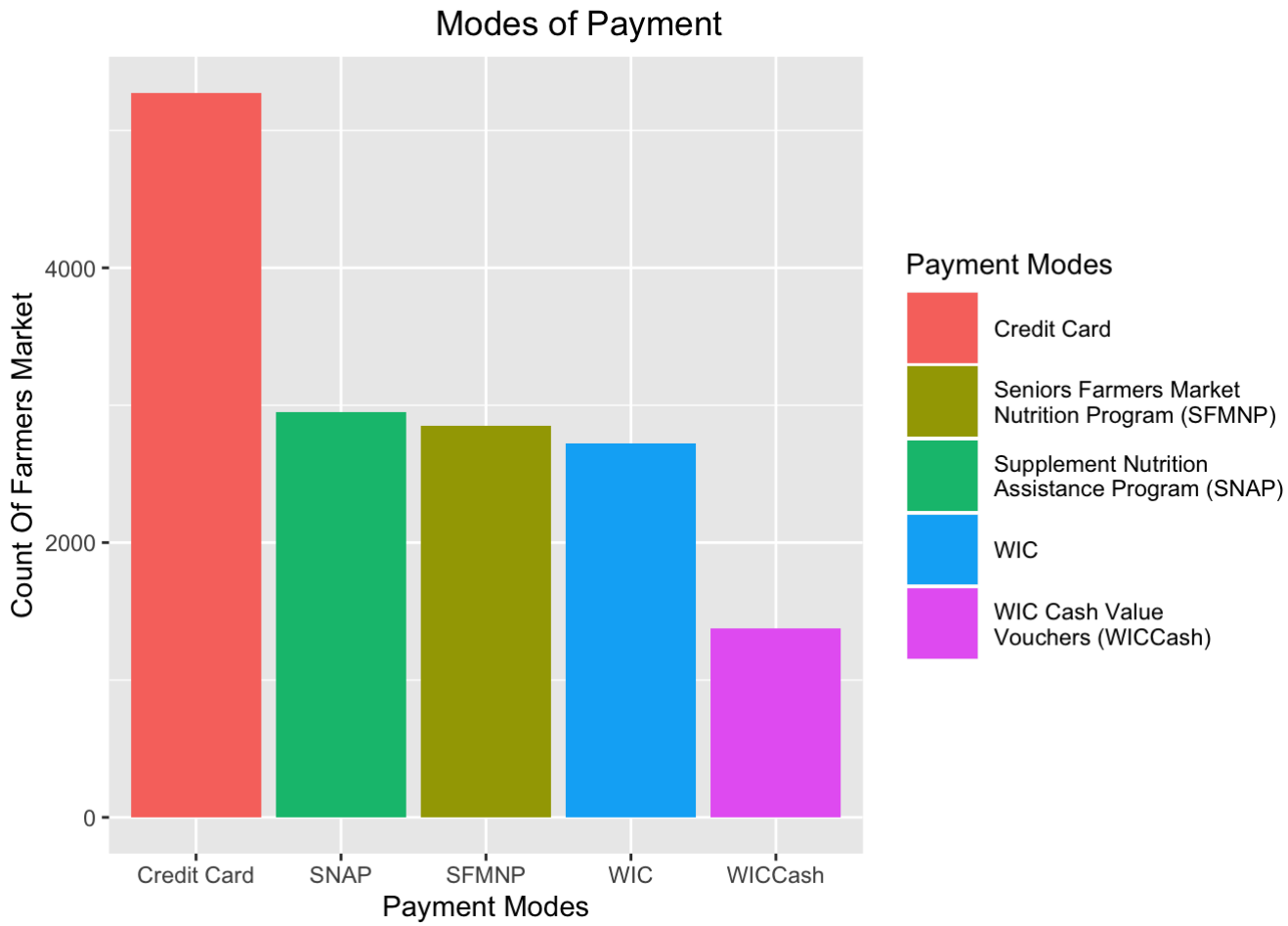
```
## Saving 7 x 5 in image
```



Payment Methods overall USA

```
#Taking count of only Y
Credit <-sum(str_count(fmarket$Credit, "Y"))
WIC    <-sum(str_count(fmarket$WIC, "Y"))
WICCash<-sum(str_count(fmarket$WICcash, "Y"))
SNAP   <-sum(str_count(fmarket$SNAP, "Y"))
SFMNP  <-sum(str_count(fmarket$SFMNP, "Y"))

#Creating new Dataframe with just the count of each method
PaymentModeDF <- data.frame(
  PaymentMode = c("Credit Card", "WIC", "WIC Cash Value Vouchers(WICCash)","Supplement N
utrition Assistance Program(SNAP)","Seniors Farmers Market Nutrition Program(SFMNP)"),
  count = c(Credit, WIC, WICCash, SNAP, SFMNP)
)

#Bar chart for the payment modes
ggplot(PaymentModeDF, aes(x=reorder(PaymentMode,count), y=count, fill=PaymentMode)) +
  geom_bar(stat = "identity") +
  scale_x_discrete(limits = c("Credit Card", "Supplement Nutrition Assistance Program(SN
AP)","Seniors Farmers Market Nutrition Program(SFMNP)","WIC", "WIC Cash Value Vouchers(W
ICCash)"),
                   labels = c("Credit Card", "SNAP", "SFMNP","WIC","WICCash")) +
  xlab("Payment Modes") +
  ylab("Count Of Farmers Market") +
  labs(fill = "Modes of payment") +
  scale_fill_discrete(name = "Payment Modes", labels = c("Credit Card","Seniors Farmers
 Market\nNutrition Program (SFMNP)","Supplement Nutrition\nAssistance Program (SNAP)","W
IC", "WIC Cash Value\nVouchers (WICCash)") ) +
  theme(legend.position = "right", plot.title = element_text(hjust = 0.7),legend.key.siz
e = unit(1, "cm")) +
  ggtitle("Modes of Payment") +
  ggsave("paymentModeVsFmarket.png", device="png", dpi = 600)
```

```
## Saving 7 x 5 in image
```

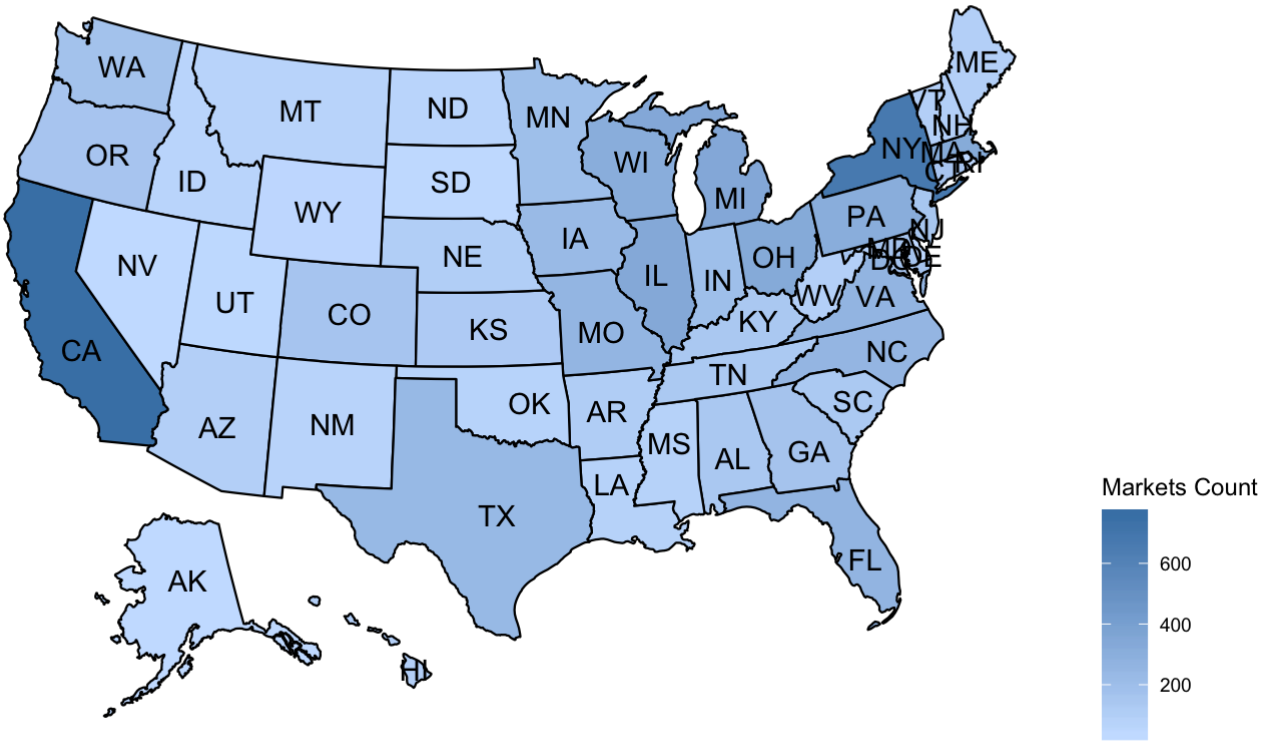


US MAP - Farmers Markets

```
#MAP PLOT FOR TOTAL COUNT OF FARMERS MARKET BY STATES IN USA
MarketCount <- fmarket %>%
  dplyr::select(State) %>%
  group_by(State)%>%
  dplyr::summarise(Fcount=n()) %>%
  arrange(desc(Fcount))

colnames(MarketCount)[1] <- "state"
#Using usmap function for ploting on US Map statewise
plot_usmap(data = MarketCount, values = "Fcount", color = "black", labels = T) +
  labs(title="Farmer Markets by States") +
  scale_fill_continuous(low = "lightsteelblue1", high = "Steelblue", name = "Markets Cou
nt") +
  theme(legend.position = "right", plot.title = element_text(size=15 , hjust = 0.7)) +
  ggsave("GeographicFmarket.png", device="png", dpi = 600)
```

```
## Saving 7 x 5 in image
```

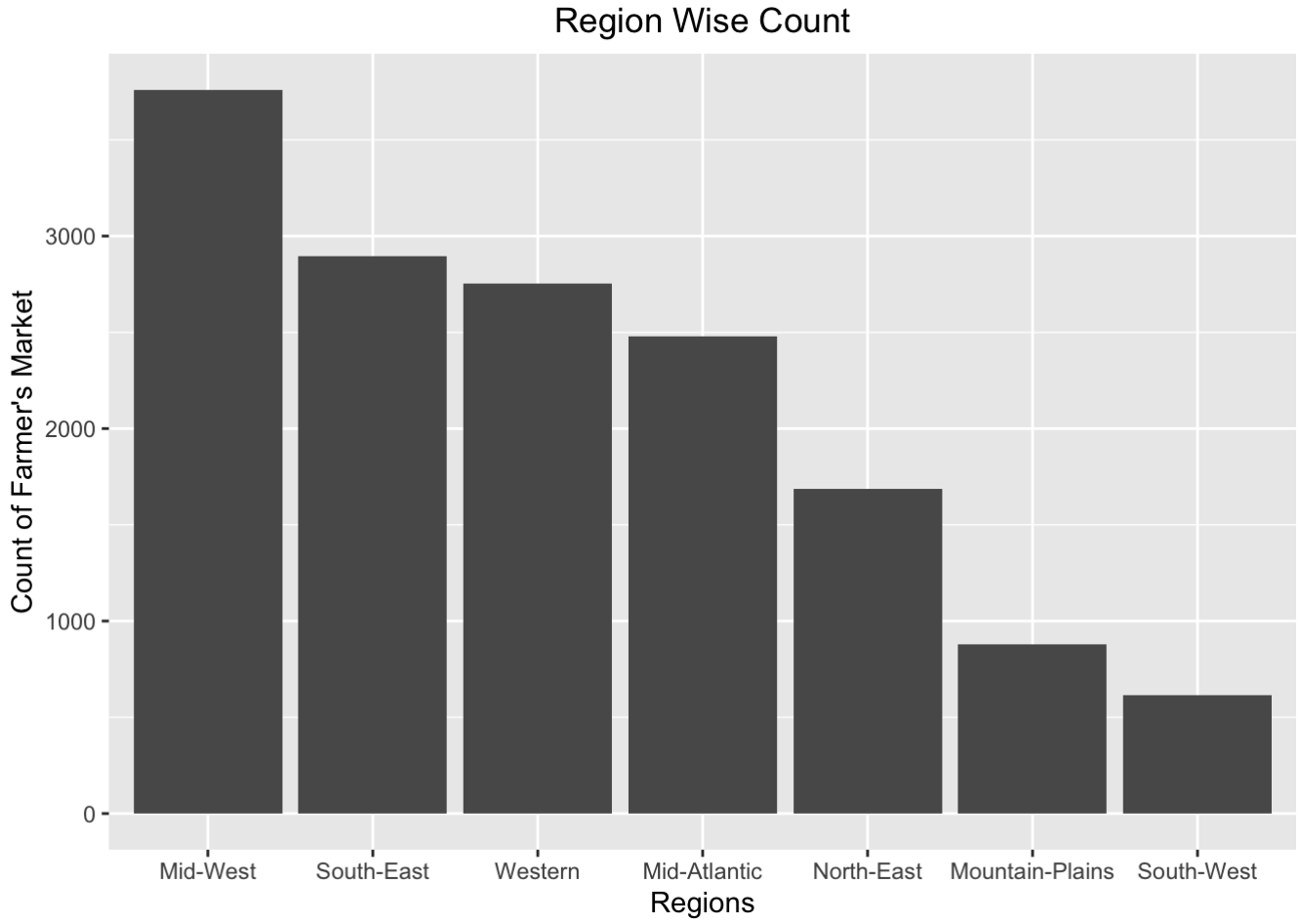
Farmer Markets by States



```
RegionGrant <-StatesInfo

#Creating regions according the one mentioned in the grant information.

MidAtlantic <-c("New Jersey","New York","Pennsylvania")
MidWest <-c("Illinois","Indiana"    ,"Michigan" ,"Ohio",    "Wisconsin",    "Iowa", "Kan
sas",   "Minnesota",    "Missouri", "North Dakota"  ,"Nebraska" ,"South Dakota")
MountainPlains<-c("Arizona","Colorado","Idaho","Montana","New Mexico","Nevada","Utah","W
yoming")
NorthEast<-c("Connecticut","Massachusetts","Maine","New Hampshire","Rhode Island","Vermo
nt")
Southeast<-c("District of Columbia","Delaware","Florida","Georgia","Maryland","North Car
olina","South Carolina","Virginia", "West Virginia","Alabama","Kentucky","Mississippi",
"Tennessee")
Southwest<-c("Arkansas","Louisiana","Oklahoma","Texas")
Western<-c("Alaska","California","Hawaii","Oregon","Washington")

#colnames(RegionGrant)[5] <- "RegionGrant"

#Assigning the new region according to the value mentioned above.
RegionGrant$NewRegion<-NA
RegionGrant[RegionGrant$State %in% MidAtlantic,5]="Mid-Atlantic"
RegionGrant[RegionGrant$State %in% MidWest,5]="Mid-West"
RegionGrant[RegionGrant$State %in% MountainPlains,5]="Mountain-Plains"
RegionGrant[RegionGrant$State %in% NorthEast,5]="North-East"
RegionGrant[RegionGrant$State %in% Southeast,5]="South-East"
RegionGrant[RegionGrant$State %in% Southwest,5]="South-West"
RegionGrant[RegionGrant$State %in% Western,5]="Western"

Fmarket_RegionGrant <-merge(y=RegionGrant,x=fmarket, by.x = "State", by.y = "State",all.
x = TRUE)

RegionGrantDF <-melt(Fmarket_RegionGrant[,c(1,2,24:28,63)], id.vars = c("FMID","NewRegio
n"))
RegionGrantDF <-RegionGrantDF[RegionGrantDF$value=='Y',]

colnames(RegionGrantDF)[3] <- "PaymentMode"

GrantRegionCount <- RegionGrantDF %>%
  group_by(PaymentMode,NewRegion)%>%
  dplyr::summarise(Gcount=n()) %>%
  arrange(desc(Gcount))

GrantRegionCount <- na.omit(GrantRegionCount)

#This colors is not working!! In some cases it worked but some cases it doesnt work
Colorvalues <- c("#999999", "#E69F00", "#56B4E9","#999999", "#E69F00", "#56B4E9", "#56B4
E9")

ggplot(GrantRegionCount, aes(x=reorder(NewRegion,-Gcount),y= Gcount)) +
  scale_fill_manual(values = Colorvalues) +
  #coord_flip() +
  geom_bar(stat="identity") +
```

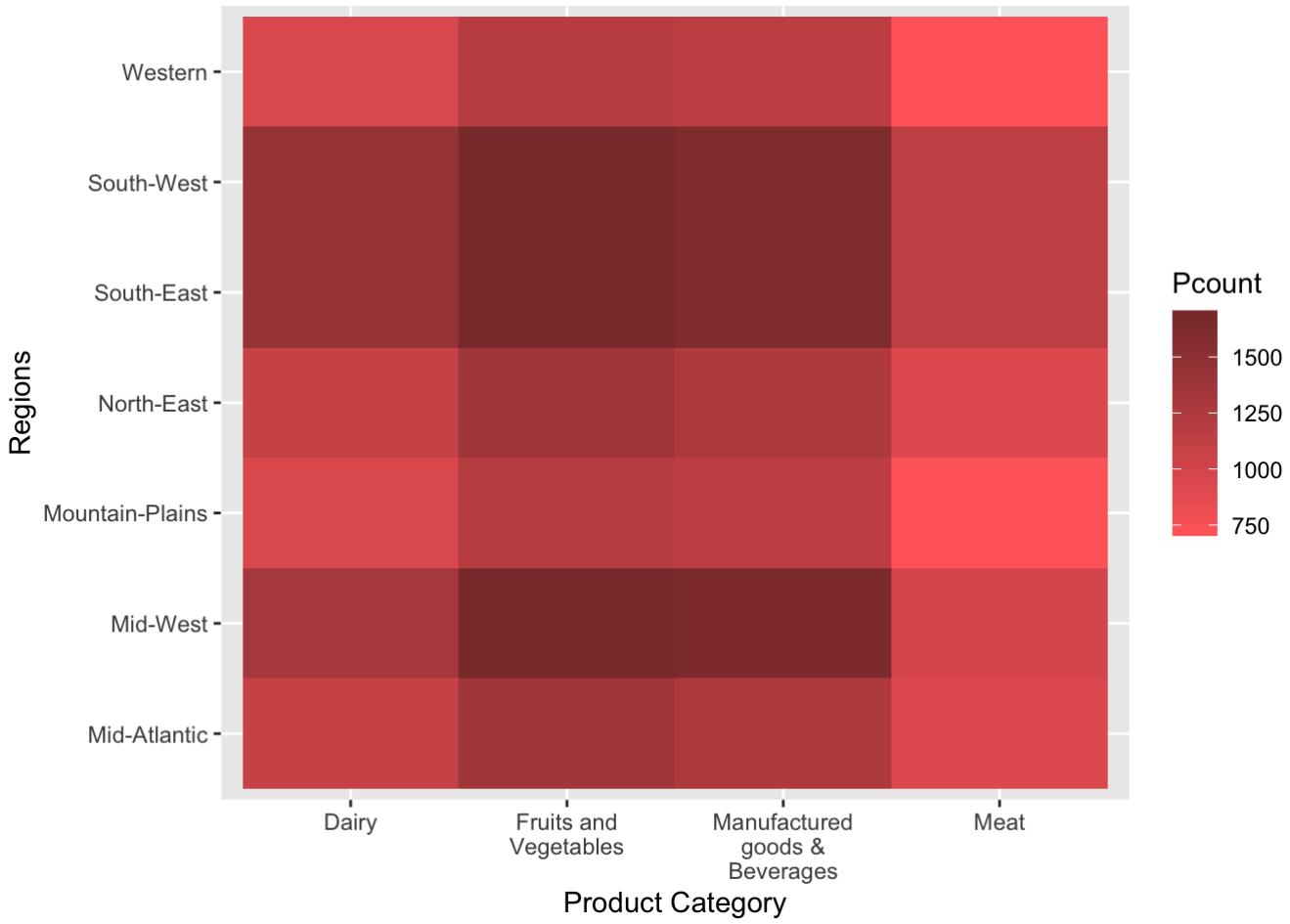
```
  labs(x = "Regions", y= "Count of Farmer's Market", title="Region Wise Count") +
  theme(plot.title = element_text(hjust = 0.5)) +
  ggsave("GrantRegion.png", device="png", dpi = 600)
```

```
## Saving 7 x 5 in image
```



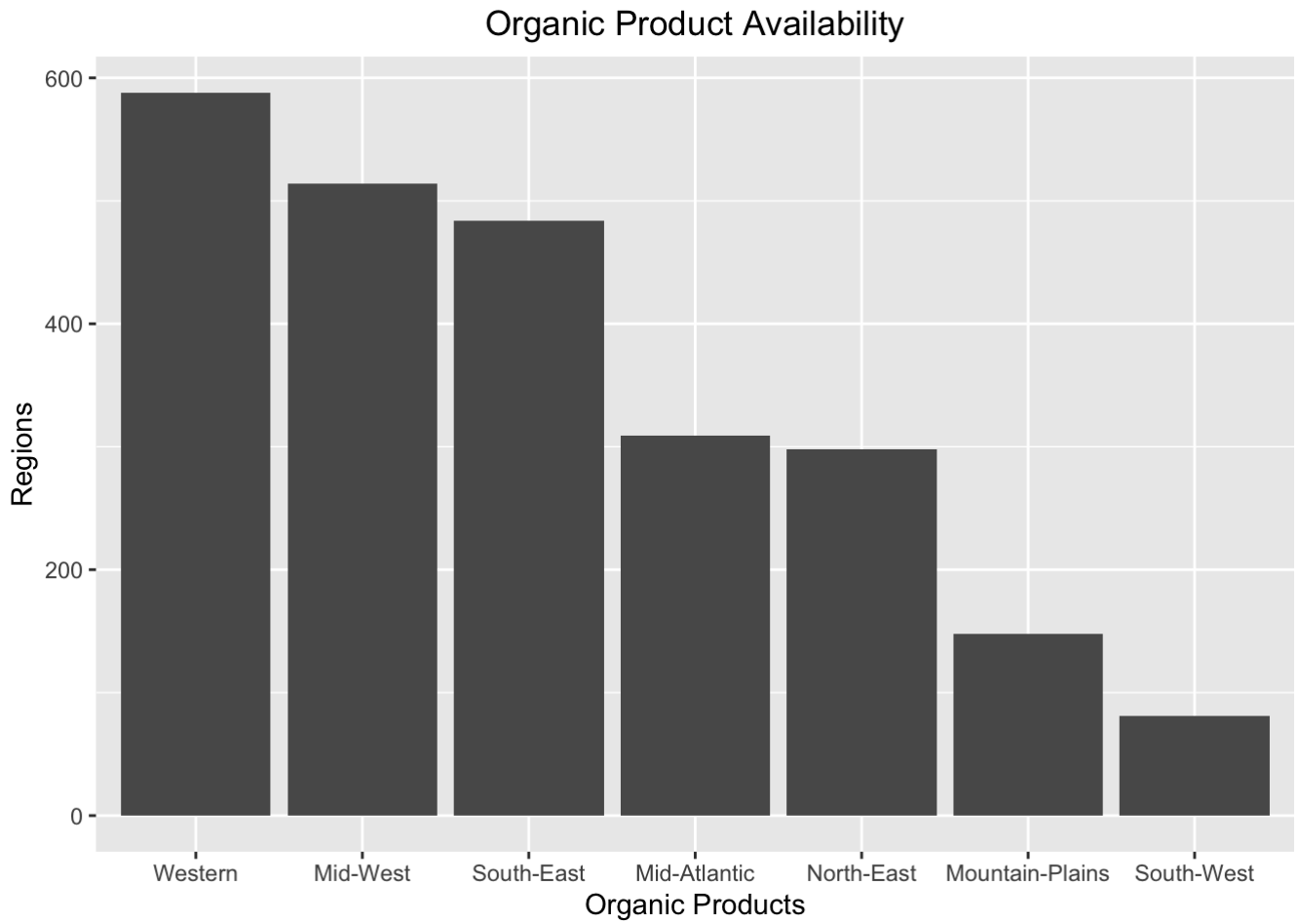

```
#Merging newregion with products
Fmarket_Product <-merge(y=RegionGrant,x=product, by.x = "Region", by.y = "Region",all.x
 = TRUE)

CategoryCount <- Fmarket_Product %>%
  group_by(NewCategory,NewRegion)%>%
  dplyr::summarise(Pcount=n_distinct(FMID)) %>%
  arrange(desc(Pcount))

#Omiting all NA's
CategoryCount <- na.omit(CategoryCount)

ggplot(CategoryCount, aes(fill=Pcount, y=NewRegion, x=NewCategory)) +
  geom_tile() +
  xlab("") +
  scale_fill_gradient(low="indianred1", high="indianred4") +
  labs(x="Product Category", y="Regions") +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  ggsave("HeatProductvsRegion.png", device="png", dpi = 600)
```

```
## Saving 7 x 5 in image
```




```
#Filtering with only Organic Y and then taken count of region.
OrganicDF <- Fmarket_RegionGrant %>% group_by(NewRegion) %>%
        dplyr::select(NewRegion, Organic)  %>%
        filter(Organic=="Y") %>%
        count(NewRegion) %>%
        arrange(desc(n))

#Omiting All NA's
OrganicDF <- na.omit(OrganicDF)

ggplot(OrganicDF, aes(y=n, x=reorder(NewRegion,-n))) +
  geom_bar(stat="identity") +
  theme(plot.title = element_text(hjust = 0.5)) +
  labs(x="Organic Products", y="Regions", title = "Organic Product Availability") +
  ggsave("Organic.png", device="png", dpi = 600)
```

```
## Saving 7 x 5 in image
```




```
#Considering only season 1 date. I have assumed the seaon 1 date to the date when the ma
rket was formed.
Fmarket_Year <-melt(Fmarket_RegionGrant[,c(1,2,13,63)], id.vars = c("FMID","NewRegion",
"State"))
Fmarket_Year <-Fmarket_Year[Fmarket_Year$value!="",]
#Seperating with the ketwork "to"
Fmarket_Year_New<-separate(data = Fmarket_Year, col = value, into = c("Start", "End"), s
ep = " to ")
```

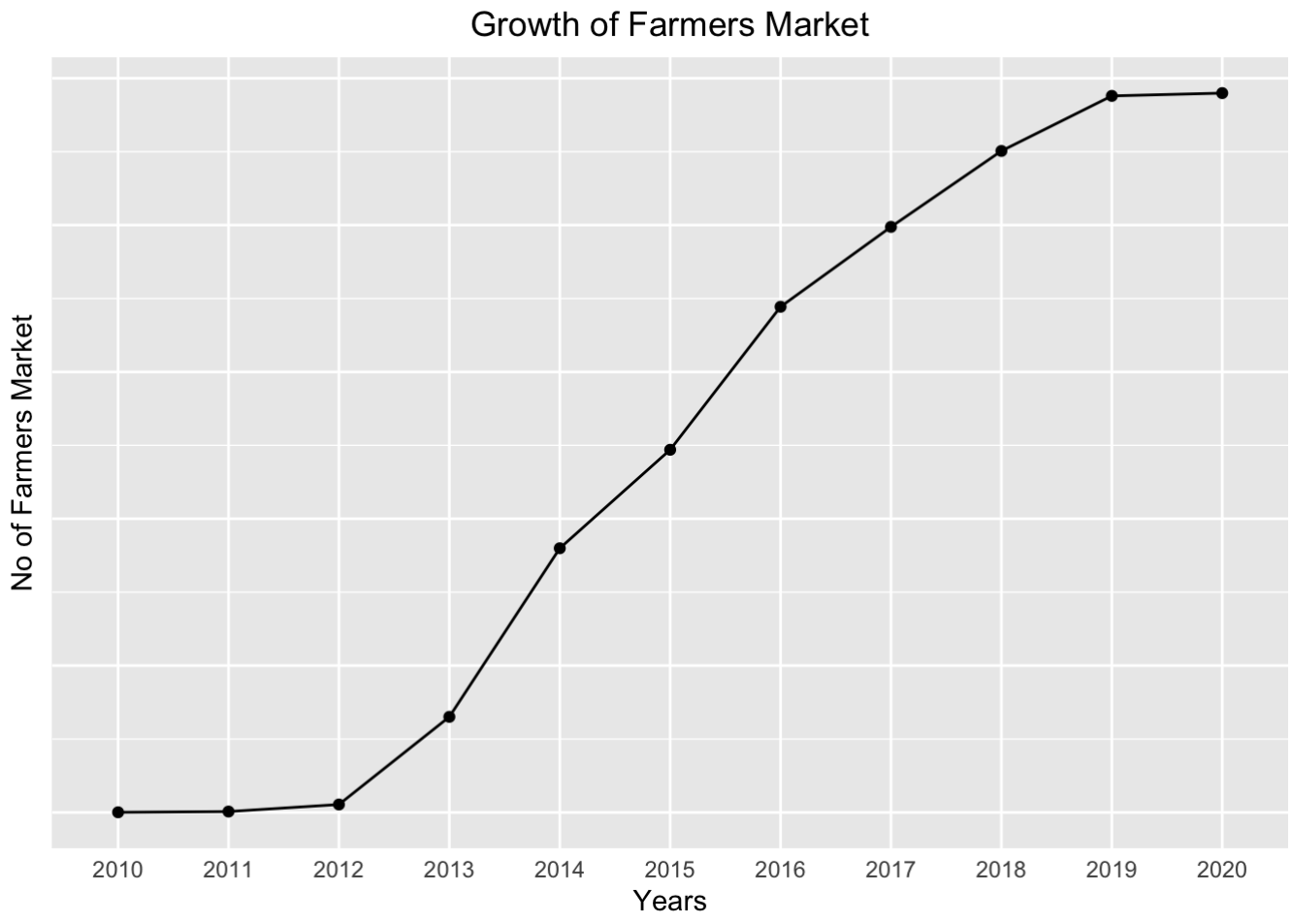
```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 24 rows [40, 121,
## 158, 160, 163, 171, 358, 596, 1008, 1310, 1313, 1315, 1321, 1348, 1375, 1478,
## 1503, 2881, 3554, 3979, ...].
```

```
#seperating the start date with / to get the different day month and year.
Fmarket_Year_New<-separate(data = Fmarket_Year_New, col = Start, into = c("Day", "Month"
, "Year"), sep = "/")
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 767 rows [7,
## 13, 28, 37, 39, 47, 52, 57, 64, 69, 71, 79, 81, 83, 101, 130, 178, 213, 223,
## 229, ...].
```

```
#Farmer Markets Across Seasons
Fmarket_Year_Group <-Fmarket_Year%>%
   group_by(variable,NewRegion)%>%
   dplyr::summarise(Count=n()) %>%
   arrange(desc(Count))

Fmarket_Year_Season1 <-Fmarket_Year_New %>%
    group_by(Year,variable)%>%
    dplyr::summarise(Count=n()) %>%
    arrange(Year)

Fmarket_Year_Season1<-Fmarket_Year_Season1[Fmarket_Year_Season1$Year!=13,]
Fmarket_Year_Season1<-Fmarket_Year_Season1[!is.na(Fmarket_Year_Season1$Year),]

Fmarket_Year_Season1[, 3] <- cumsum(Fmarket_Year_Season1[, 3])

ggplot(Fmarket_Year_Season1,aes(y=Count, x=(Year),group=variable))+
  geom_line()+
  geom_point()+
  labs(x="Years", y="No of Farmers Market",title="Growth of Farmers Market")+
  theme(axis.text.y = element_blank(),axis.ticks = element_blank(), plot.title = element
_text(hjust = 0.5)) +
  ggsave("TotalGrowth.png", device="png", dpi = 600)
```

```
## Saving 7 x 5 in image
```



```
#In this chunk all the computation is done for all the regions seperatly and then arange
d it using ggarrange.

#This if for specifically MidWest Region
Fmarket_Year_Region1 <-Fmarket_Year_New %>%
  filter(NewRegion == "Mid-West")

Fmarket_Year_Region1 <-Fmarket_Year_Region1 %>%
    group_by(Year,variable)%>%
    dplyr::summarise(Count=n()) %>%
    arrange(Year)

Fmarket_Year_Region1<-Fmarket_Year_Region1[!is.na(Fmarket_Year_Region1$Year),]
Fmarket_Year_Region1<-Fmarket_Year_Region1[Fmarket_Year_Region1$Year!=13,]


Fmarket_Year_Region1[, 3] <- cumsum(Fmarket_Year_Region1[, 3])

R1 <- ggplot(Fmarket_Year_Region1,aes(y=Count, x=(Year),group=variable))+
  geom_line()+
  geom_point()+
  labs(title="Mid West")+
  theme(axis.text.y = element_blank(),axis.ticks = element_blank())

#This if for specifically South-East Region

Fmarket_Year_Region2 <-Fmarket_Year_New %>%
  filter(NewRegion == "South-East")

Fmarket_Year_Region2 <-Fmarket_Year_Region2 %>%
    group_by(Year,variable)%>%
    dplyr::summarise(Count=n()) %>%
    arrange(Year)

Fmarket_Year_Region2<-Fmarket_Year_Region2[!is.na(Fmarket_Year_Region2$Year),]
Fmarket_Year_Region2<-Fmarket_Year_Region2[Fmarket_Year_Region2$Year!=13,]


Fmarket_Year_Region2[, 3] <- cumsum(Fmarket_Year_Region2[, 3])

R2 <- ggplot(Fmarket_Year_Region2,aes(y=Count, x=(Year),group=variable))+
  geom_line()+
  geom_point()+
  labs(title="South-East")+
  theme(axis.text.y = element_blank(),axis.ticks = element_blank())

#This if for specifically Mid-Atlantic Region

Fmarket_Year_Region3 <-Fmarket_Year_New %>%
  filter(NewRegion == "Mid-Atlantic")

Fmarket_Year_Region3 <-Fmarket_Year_Region3 %>%
    group_by(Year,variable)%>%
```

```
    dplyr::summarise(Count=n()) %>%
    arrange(Year)

Fmarket_Year_Region3<-Fmarket_Year_Region3[!is.na(Fmarket_Year_Region3$Year),]
Fmarket_Year_Region3<-Fmarket_Year_Region3[Fmarket_Year_Region3$Year!=13,]


Fmarket_Year_Region3[, 3] <- cumsum(Fmarket_Year_Region3[, 3])

R3 <- ggplot(Fmarket_Year_Region3,aes(y=Count, x=(Year),group=variable))+
  geom_line()+
  geom_point()+
  labs(title="Mid-Atlantic")+
  theme(axis.text.y = element_blank(),axis.ticks = element_blank())

#This if for specifically North-East Region

Fmarket_Year_Region4 <-Fmarket_Year_New %>%
  filter(NewRegion == "North-East")

Fmarket_Year_Region4 <-Fmarket_Year_Region4 %>%
    group_by(Year,variable)%>%
    dplyr::summarise(Count=n()) %>%
    arrange(Year)

Fmarket_Year_Region4<-Fmarket_Year_Region4[!is.na(Fmarket_Year_Region4$Year),]
Fmarket_Year_Region4<-Fmarket_Year_Region4[Fmarket_Year_Region4$Year!=13,]


Fmarket_Year_Region4[, 3] <- cumsum(Fmarket_Year_Region4[, 3])

R4 <- ggplot(Fmarket_Year_Region4,aes(y=Count, x=(Year),group=variable))+
  geom_line()+
  geom_point()+
  labs(title="North-East")+
  theme(axis.text.y = element_blank(),axis.ticks = element_blank())

#This if for specifically Mountain-Plains Region

Fmarket_Year_Region5 <-Fmarket_Year_New %>%
  filter(NewRegion == "Mountain-Plains")

Fmarket_Year_Region5 <-Fmarket_Year_Region5 %>%
    group_by(Year,variable)%>%
    dplyr::summarise(Count=n()) %>%
    arrange(Year)

Fmarket_Year_Region5<-Fmarket_Year_Region5[!is.na(Fmarket_Year_Region5$Year),]
Fmarket_Year_Region5<-Fmarket_Year_Region5[Fmarket_Year_Region5$Year!=13,]


Fmarket_Year_Region5[, 3] <- cumsum(Fmarket_Year_Region5[, 3])

R5 <- ggplot(Fmarket_Year_Region5,aes(y=Count, x=(Year),group=variable))+
```

```
  geom_line()+
  geom_point()+
  labs(title="Mountain-Plains")+
  theme(axis.text.y = element_blank(),axis.ticks = element_blank())

#This if for specifically South-West Region

Fmarket_Year_Region6 <-Fmarket_Year_New %>%
  filter(NewRegion == "South-West")

Fmarket_Year_Region6 <-Fmarket_Year_Region6 %>%
    group_by(Year,variable)%>%
    dplyr::summarise(Count=n()) %>%
    arrange(Year)

Fmarket_Year_Region6<-Fmarket_Year_Region6[!is.na(Fmarket_Year_Region6$Year),]
Fmarket_Year_Region6<-Fmarket_Year_Region6[Fmarket_Year_Region6$Year!=13,]


Fmarket_Year_Region6[, 3] <- cumsum(Fmarket_Year_Region6[, 3])

R6 <- ggplot(Fmarket_Year_Region6,aes(y=Count, x=(Year),group=variable))+
  geom_line()+
  geom_point()+
  labs(title="South-West")+
  theme(axis.text.y = element_blank(),axis.ticks = element_blank())

#This if for specifically Western Region

Fmarket_Year_Region7 <-Fmarket_Year_New %>%
  filter(NewRegion == "Western")

Fmarket_Year_Region7 <-Fmarket_Year_Region7 %>%
    group_by(Year,variable)%>%
    dplyr::summarise(Count=n()) %>%
    arrange(Year)

Fmarket_Year_Region7<-Fmarket_Year_Region7[!is.na(Fmarket_Year_Region7$Year),]
Fmarket_Year_Region7<-Fmarket_Year_Region7[Fmarket_Year_Region7$Year!=13,]


Fmarket_Year_Region7[, 3] <- cumsum(Fmarket_Year_Region7[, 3])

R7 <- ggplot(Fmarket_Year_Region7,aes(y=Count, x=(Year),group=variable))+
  geom_line()+
  geom_point()+
  labs(title="Western")+
  theme(axis.text.y = element_blank(),axis.ticks = element_blank())

ggpubr::ggarrange(R1,R2,R3,R4,R5,R6,NA,R7) +
  labs(title="\nGrowth of Farmers Market for various Regions\n") +
  theme(axis.text.y = element_blank(),axis.ticks = element_blank(), plot.title = element
_text(hjust = 0.5)) +
  ggsave("AllRegionsGrowth.png", device="png", dpi = 600)
```

```
## Warning in as_grob.default(plot): Cannot convert object of class logical into a
## grob.
```

```
## Saving 7 x 5 in image
```