Adidas\_sales

Joshua

2023-06-06

**About Adidas**

Adidas AG, the Adidas group, is one of the largest manufacturers of sports shoes and clothing in Europe and the second-biggest in the world after the main competitor Nike. Adidas produces and sells sports clothing, sneakers, running shoes, bags, accessories like watches and sunglasses, and so on through its e-com shop, over a thousand retail stores, and several franchises. The brand is headquartered in Herzogenaurach, Germany.

Data set was download from github account

# load the packages  
  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.2.0  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(readr)  
library(dplyr)  
library(ggplot2)  
library(janitor)

##   
## Attaching package: 'janitor'  
##   
## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

library(readxl)

#load the dataset  
df <- read\_xlsx("Adidas Sales (2020-2021).xlsx")

Data Wrangling and cleaning

head(df)

## # A tibble: 6 × 12  
## Retailer Retai…¹ Invoi…² Region State City Product Price…³ Units…⁴ Total…⁵  
## <chr> <dbl> <dbl> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Foot Locker 1185732 43831 North… New … New … Men's … 50 1200 60000  
## 2 Foot Locker 1185732 43832 North… New … New … Men's … 50 1000 50000  
## 3 Foot Locker 1185732 43833 North… New … New … Women'… 40 1000 40000  
## 4 Foot Locker 1185732 43834 North… New … New … Women'… 45 850 38250  
## 5 Foot Locker 1185732 43835 North… New … New … Men's … 60 900 54000  
## 6 Foot Locker 1185732 43836 North… New … New … Women'… 50 1000 50000  
## # … with 2 more variables: `Operating Profit` <dbl>, `Sales Method` <chr>, and  
## # abbreviated variable names ¹​`Retailer ID`, ²​`Invoice Date`,  
## # ³​`Price per Unit`, ⁴​`Units Sold`, ⁵​`Total Sales`

summary(df)

## Retailer Retailer ID Invoice Date Region   
## Length:9648 Min. :1128299 Min. :43831 Length:9648   
## Class :character 1st Qu.:1185732 1st Qu.:44244 Class :character   
## Mode :character Median :1185732 Median :44351 Mode :character   
## Mean :1173850 Mean :44327   
## 3rd Qu.:1185732 3rd Qu.:44455   
## Max. :1197831 Max. :44561   
## State City Product Price per Unit   
## Length:9648 Length:9648 Length:9648 Min. : 7.00   
## Class :character Class :character Class :character 1st Qu.: 35.00   
## Mode :character Mode :character Mode :character Median : 45.00   
## Mean : 45.22   
## 3rd Qu.: 55.00   
## Max. :110.00   
## Units Sold Total Sales Operating Profit Sales Method   
## Min. : 0.0 Min. : 0.0 Min. : 0.0 Length:9648   
## 1st Qu.: 106.0 1st Qu.: 425.4 1st Qu.: 192.2 Class :character   
## Median : 176.0 Median : 957.6 Median : 437.1 Mode :character   
## Mean : 256.9 Mean : 9327.3 Mean : 3442.5   
## 3rd Qu.: 350.0 3rd Qu.:15000.0 3rd Qu.: 5206.2   
## Max. :1275.0 Max. :82500.0 Max. :39000.0

summary(df$`Price per Unit`)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 7.00 35.00 45.00 45.22 55.00 110.00

#count the null value  
sum(is.na(df))

## [1] 0

unique(count(df))

## # A tibble: 1 × 1  
## n  
## <int>  
## 1 9648

#   
sum(duplicated(df))

## [1] 0

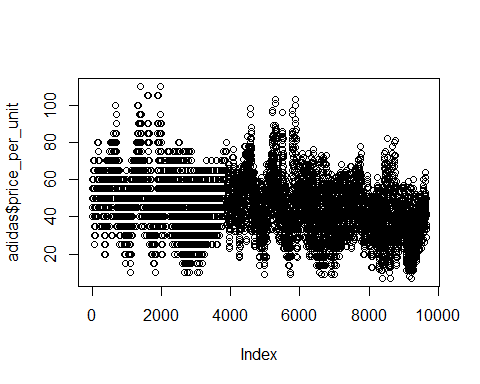
#cleaning the column  
adidas <- df %>% clean\_names()

str(adidas)

## tibble [9,648 × 12] (S3: tbl\_df/tbl/data.frame)  
## $ retailer : chr [1:9648] "Foot Locker" "Foot Locker" "Foot Locker" "Foot Locker" ...  
## $ retailer\_id : num [1:9648] 1185732 1185732 1185732 1185732 1185732 ...  
## $ invoice\_date : num [1:9648] 43831 43832 43833 43834 43835 ...  
## $ region : chr [1:9648] "Northeast" "Northeast" "Northeast" "Northeast" ...  
## $ state : chr [1:9648] "New York" "New York" "New York" "New York" ...  
## $ city : chr [1:9648] "New York" "New York" "New York" "New York" ...  
## $ product : chr [1:9648] "Men's Street Footwear" "Men's Athletic Footwear" "Women's Street Footwear" "Women's Athletic Footwear" ...  
## $ price\_per\_unit : num [1:9648] 50 50 40 45 60 50 50 50 40 45 ...  
## $ units\_sold : num [1:9648] 1200 1000 1000 850 900 1000 1250 900 950 825 ...  
## $ total\_sales : num [1:9648] 60000 50000 40000 38250 54000 ...  
## $ operating\_profit: num [1:9648] 30000 15000 14000 13388 16200 ...  
## $ sales\_method : chr [1:9648] "In-store" "In-store" "In-store" "In-store" ...

**Now we can analyze the data**

plot(adidas$price\_per\_unit)

 In this Region column we have a problem of the West and West one starting with capital letter and the other with lowercase case So let’s check

adidas %>% count(region)

## # A tibble: 6 × 2  
## region n  
## <chr> <int>  
## 1 Midwest 1872  
## 2 Northeast 2376  
## 3 South 1728  
## 4 Southeast 1224  
## 5 West 2447  
## 6 west 1

The same with products column

adidas %>% count(product)

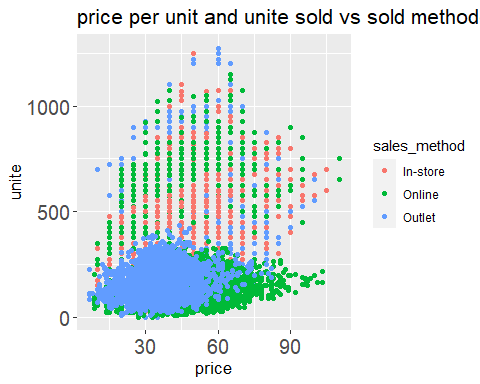
## # A tibble: 7 × 2  
## product n  
## <chr> <int>  
## 1 Men's Apparel 1605  
## 2 Men's Athletic Footwear 1610  
## 3 Men's Street Footwear 1610  
## 4 Men's aparel 1  
## 5 Women's Apparel 1608  
## 6 Women's Athletic Footwear 1606  
## 7 Women's Street Footwear 1608

Men Apparel

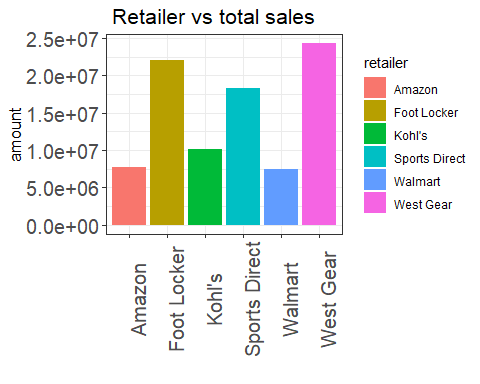
Let’s clean that

# Replace String with Another String on a single column  
  
adidas$region [adidas$region == "west"] <- "West"  
adidas$product [adidas$product == "Men's aparel"] <- "Men's Apparel"

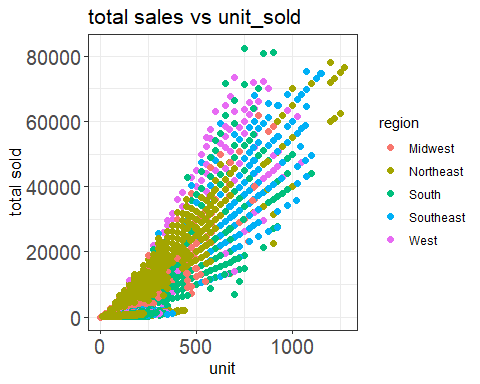
ggplot(adidas, aes(x=price\_per\_unit, y=units\_sold )) +   
 geom\_point(aes(color=sales\_method )) +   
 labs(y="unite",   
 x="price",   
 title="price per unit and unite sold vs sold methode ")+   
 theme\_gray()+ # Default theme   
 theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=15),  
 axis.text.y= element\_text(size=15), axis.title=element\_text(size=13))



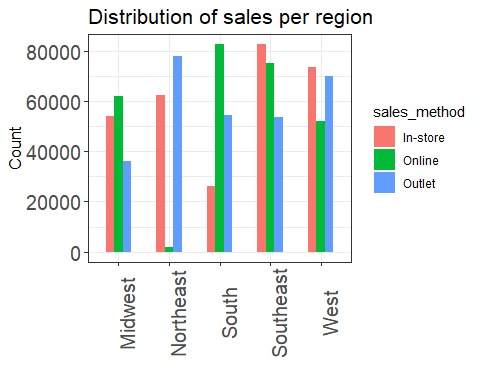
ggplot(adidas, aes( x =retailer, y= total\_sales)) +   
 geom\_col(aes(fill=retailer )) +   
 labs(x=" ",   
 y="amount",  
 title=" Retailer vs total sales")+   
 theme\_bw()+  
theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=15,angle=90),  
 axis.text.y= element\_text(size=15), axis.title=element\_text(size=13))



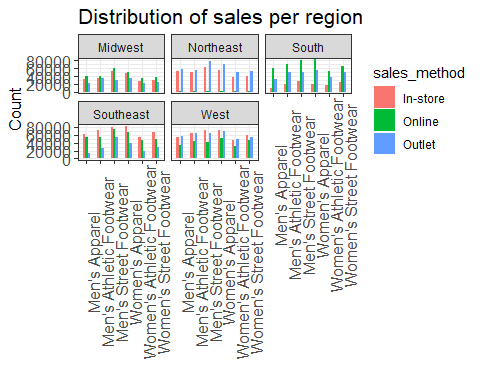
ggplot(adidas, aes(x=units\_sold ,y=total\_sales, color=region)) +  
 geom\_jitter(width = .5, size=2)+ # Adjusting size and width of points  
 labs(x="unit",  
 y="total sold",   
 title="total sales vs unit\_sold ")+   
theme\_bw()+  
theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=15),  
 axis.text.y= element\_text(size=15), axis.title=element\_text(size=13))



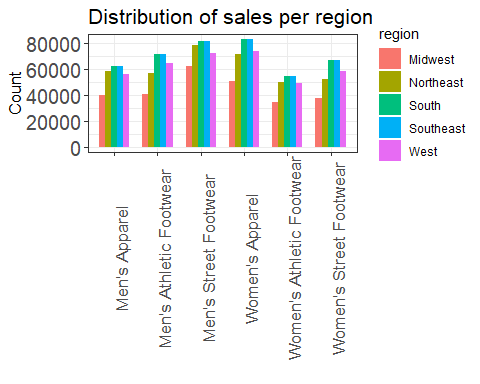
ggplot(adidas, aes(x= region, y=total\_sales, fill=sales\_method))+  
geom\_col(position='dodge', width = 0.5)+ # Stack for stacked chart  
 labs(x=" ",  
 y="Count",   
 title="Distribution of sales per region")+   
theme\_bw()+  
theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=15,angle=90),  
 axis.text.y= element\_text(size=15), axis.title=element\_text(size=13))



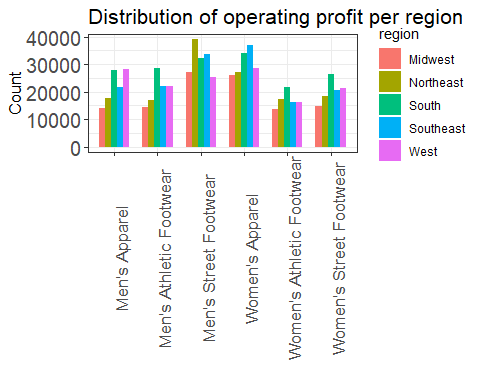
ggplot(adidas, aes(x= product, y=total\_sales, fill=sales\_method))+  
geom\_col(position='dodge', width = 0.5)+ # Stack for stacked chart  
 labs(x=" ",  
 y="Count",   
 title="Distribution of sales per region")+ facet\_wrap(~region) +  
theme\_bw()+  
theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=12,angle=90),  
 axis.text.y= element\_text(size=12), axis.title=element\_text(size=13))



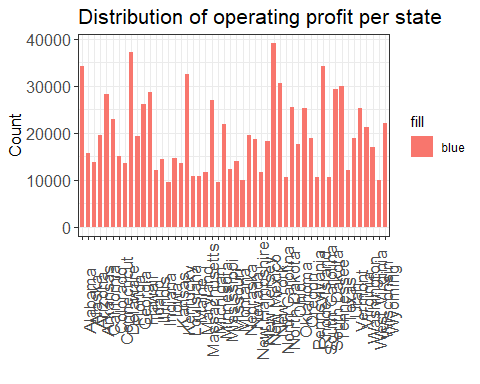
ggplot(adidas, aes(x= product, y=total\_sales, fill=region))+  
geom\_col(position='dodge', width = 0.7)+ # Stack for stacked chart  
 labs(x=" ",  
 y="Count",   
 title="Distribution of sales per region")+   
theme\_bw()+  
theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=13,angle=90),  
 axis.text.y= element\_text(size=15), axis.title=element\_text(size=13))



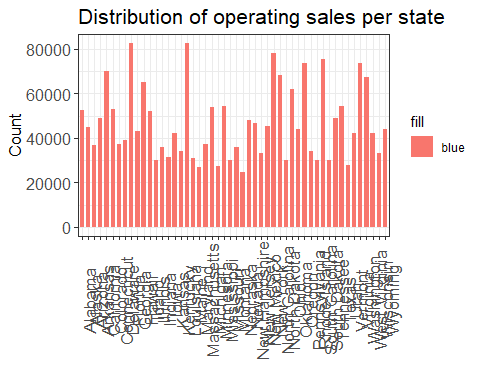
ggplot(adidas, aes(x= product, y=operating\_profit, fill=region))+  
geom\_col(position='dodge', width = 0.7)+ # Stack for stacked chart  
 labs(x=" ",  
 y="Count",   
 title="Distribution of operating profit per region")+   
theme\_bw()+  
theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=13,angle=90),  
 axis.text.y= element\_text(size=15), axis.title=element\_text(size=13))



ggplot(adidas, aes(x= state, y=operating\_profit, fill="blue"))+  
geom\_col(position='dodge', width = 0.7)+ # Stack for stacked chart  
 labs(x=" ",  
 y="Count",   
 title="Distribution of operating profit per state")+   
theme\_bw()+  
theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=13,angle=90),  
 axis.text.y= element\_text(size=12), axis.title=element\_text(size=13))



ggplot(adidas, aes(x= state, y=total\_sales, fill="blue"))+  
geom\_col(position='dodge', width = 0.7)+ # Stack for stacked chart  
 labs(x=" ",  
 y="Count",   
 title="Distribution of operating sales per state")+   
theme\_bw()+  
theme(plot.title = element\_text(size=16),axis.text.x= element\_text(size=13,angle=90),  
 axis.text.y= element\_text(size=12), axis.title=element\_text(size=13))



adidas %>% group\_by(region,retailer) %>%summarise(sum(total\_sales))

## `summarise()` has grouped output by 'region'. You can override using the  
## `.groups` argument.

## # A tibble: 28 × 3  
## # Groups: region [5]  
## region retailer `sum(total\_sales)`  
## <chr> <chr> <dbl>  
## 1 Midwest Amazon 1683587.  
## 2 Midwest Foot Locker 4798739.  
## 3 Midwest Kohl's 2222942.  
## 4 Midwest Sports Direct 2620719.  
## 5 Midwest West Gear 2254059.  
## 6 Northeast Amazon 3626259   
## 7 Northeast Foot Locker 6532647.  
## 8 Northeast Kohl's 1403117.  
## 9 Northeast Sports Direct 2469810.  
## 10 Northeast Walmart 1371200.  
## # … with 18 more rows

adidas %>% group\_by(region,product) %>%summarise(sum(total\_sales))

## `summarise()` has grouped output by 'region'. You can override using the  
## `.groups` argument.

## # A tibble: 30 × 3  
## # Groups: region [5]  
## region product `sum(total\_sales)`  
## <chr> <chr> <dbl>  
## 1 Midwest Men's Apparel 1812566.  
## 2 Midwest Men's Athletic Footwear 2130554.  
## 3 Midwest Men's Street Footwear 3832281   
## 4 Midwest Women's Apparel 2820638.  
## 5 Midwest Women's Athletic Footwear 1359517.  
## 6 Midwest Women's Street Footwear 1624490.  
## 7 Northeast Men's Apparel 2574441.  
## 8 Northeast Men's Athletic Footwear 2887424.  
## 9 Northeast Men's Street Footwear 5102502.  
## 10 Northeast Women's Apparel 3754308.  
## # … with 20 more rows

adidas %>% group\_by(region,product) %>%summarise(sum(operating\_profit))

## `summarise()` has grouped output by 'region'. You can override using the  
## `.groups` argument.

## # A tibble: 30 × 3  
## # Groups: region [5]  
## region product `sum(operating\_profit)`  
## <chr> <chr> <dbl>  
## 1 Midwest Men's Apparel 723253.  
## 2 Midwest Men's Athletic Footwear 740681.  
## 3 Midwest Men's Street Footwear 1499737.  
## 4 Midwest Women's Apparel 1232508.  
## 5 Midwest Women's Athletic Footwear 504387.  
## 6 Midwest Women's Street Footwear 580568.  
## 7 Northeast Men's Apparel 816154.  
## 8 Northeast Men's Athletic Footwear 1025067.  
## 9 Northeast Men's Street Footwear 2135632.  
## 10 Northeast Women's Apparel 1349922.  
## # … with 20 more rows

adidas %>% group\_by(city) %>%summarise(sum(total\_sales))

## # A tibble: 52 × 2  
## city `sum(total\_sales)`  
## <chr> <dbl>  
## 1 Albany 2442780.  
## 2 Albuquerque 1986502.  
## 3 Anchorage 1475310.  
## 4 Atlanta 1899747.  
## 5 Baltimore 775806.  
## 6 Billings 1571089.  
## 7 Birmingham 1763342.  
## 8 Boise 1927688.  
## 9 Boston 1107681   
## 10 Burlington 1435292.  
## # … with 42 more rows

adidas %>% group\_by(city) %>%summarise(sum(operating\_profit))

## # A tibble: 52 × 2  
## city `sum(operating\_profit)`  
## <chr> <dbl>  
## 1 Albany 942985.  
## 2 Albuquerque 673806.  
## 3 Anchorage 449835.  
## 4 Atlanta 689329.  
## 5 Baltimore 275764.  
## 6 Billings 623204.  
## 7 Birmingham 914758.  
## 8 Boise 812112.  
## 9 Boston 335388.  
## 10 Burlington 578597.  
## # … with 42 more rows

…..