Homework Data Viz

Bowornrat.K

2024-08-26

# **Homework\_Data\_Viz\_Batch10**

## *Packages tidyverse*

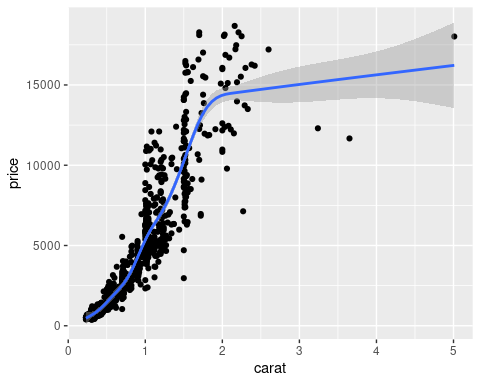
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

## *Ramdom a sample and find a relationship between carat and price*

set.seed(52)  
small\_diamons <- sample\_n(diamonds, 1000)  
  
ggplot(small\_diamons, aes(carat, price)) +  
 geom\_point() +  
 geom\_smooth()

## `geom\_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'

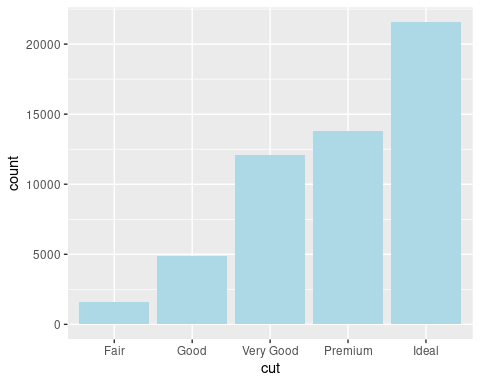


## *box plot by group*

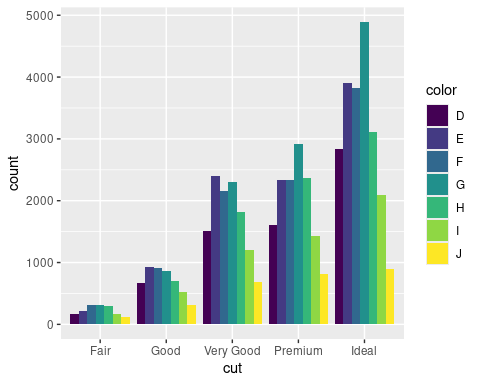
diamonds %>%  
 count(cut)

## # A tibble: 5 × 2  
## cut n  
## <ord> <int>  
## 1 Fair 1610  
## 2 Good 4906  
## 3 Very Good 12082  
## 4 Premium 13791  
## 5 Ideal 21551

ggplot(diamonds, aes(cut)) +  
 geom\_bar(fill = "lightblue")



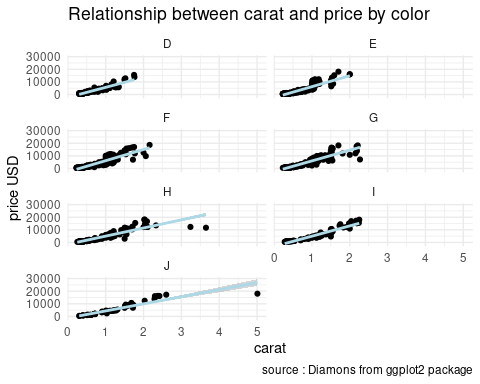
ggplot(diamonds, aes(cut, fill = color)) +  
 geom\_bar(position = "dodge")



## *Relationship between carat and price by color in facet*

ggplot(small\_diamons, aes(carat, price)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", col="lightblue") +  
 facet\_wrap(~color, ncol = 2) +  
 theme\_minimal() +  
 labs(title = "Relationship between carat and price by color",  
 x = "carat",  
 y = "price USD",  
 caption = "source : Diamons from ggplot2 package")

## `geom\_smooth()` using formula = 'y ~ x'



## *relationship between cut and price*

set.seed(52)  
small\_diamons <- sample\_n(diamonds, 100)  
ggplot(small\_diamons, aes(price, cut)) +  
 geom\_point() +  
 geom\_smooth() +  
 theme\_minimal()

## `geom\_smooth()` using method = 'loess' and formula = 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
## : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
## : pseudoinverse used at 585.95

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
## : neighborhood radius 2378.1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
## : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = parametric,  
## : There are other near singularities as well. 67109

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
## else if (is.data.frame(newdata))  
## as.matrix(model.frame(delete.response(terms(object)), : span too small. fewer  
## data values than degrees of freedom.

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
## else if (is.data.frame(newdata))  
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at  
## 585.95

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
## else if (is.data.frame(newdata))  
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius  
## 2378.1

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
## else if (is.data.frame(newdata))  
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition  
## number 0

## Warning in predLoess(object$y, object$x, newx = if (is.null(newdata)) object$x  
## else if (is.data.frame(newdata))  
## as.matrix(model.frame(delete.response(terms(object)), : There are other near  
## singularities as well. 67109

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning  
## -Inf

