Chapter11 Modelling for Visualisation

## 11.1 Introduction

reference : Wickham H, Dianne C, Heike H (2015) Visualizing statistical models: removing the blindfold. Stat Anal Data Min: ASA Data Sci J 8(4):203–25

library(ggplot2)  
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(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

## 11.2 Removing Trend

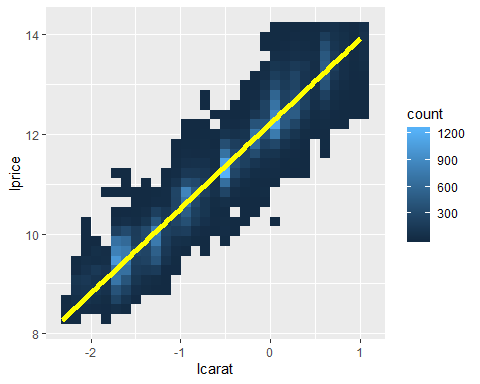
Diamonds data analysis!!

powerful relationship between size and price -> difficult to see the impact of cut, colour and clarity

diamonds2 <- diamonds %>%  
 filter(carat <= 2) %>%  
 mutate(  
 lcarat = log2(carat),  
 lprice = log2(price)  
 )  
  
head(diamonds2)

## # A tibble: 6 x 12  
## carat cut color clarity depth table price x y z lcarat lprice  
## <dbl> <ord> <ord> <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43 -2.12 8.35  
## 2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31 -2.25 8.35  
## 3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31 -2.12 8.35  
## 4 0.290 Premium I VS2 62.4 58 334 4.2 4.23 2.63 -1.79 8.38  
## 5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75 -1.69 8.39  
## 6 0.24 Very Go… J VVS2 62.8 57 336 3.94 3.96 2.48 -2.06 8.39

ggplot(diamonds2, aes(lcarat, lprice)) +  
 geom\_bin2d() +  
 geom\_smooth(method = "lm", se = FALSE, size = 2, colour = "yellow")

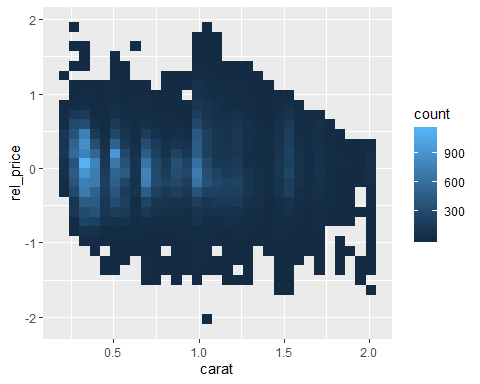


mod = lm(lprice ~ lcarat, data = diamonds2)  
coef(summary(mod))

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 12.20892 0.002108989 5788.9930 0  
## lcarat 1.69659 0.002078441 816.2798 0

Using residuals!!

diamonds2 <- diamonds2 %>% mutate(rel\_price = resid(mod))  
  
ggplot(diamonds2, aes(carat, rel\_price)) +  
 geom\_bin2d()

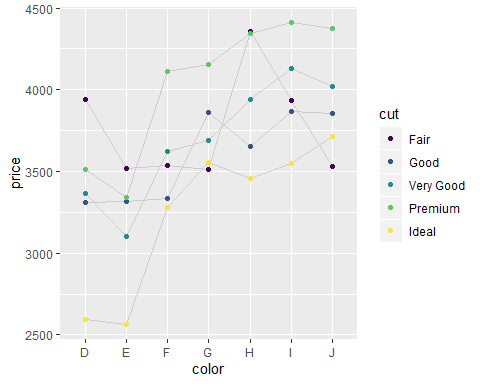


how color and cut affect the value of a diamond!

color\_cut <- diamonds2 %>%  
 group\_by(color, cut) %>%  
 summarise(  
 price = mean(price),  
 rel\_price = mean(rel\_price)  
 )  
head(color\_cut)

## # A tibble: 6 x 4  
## # Groups: color [2]  
## color cut price rel\_price  
## <ord> <ord> <dbl> <dbl>  
## 1 D Fair 3939. -0.0755  
## 2 D Good 3309. -0.0472  
## 3 D Very Good 3368. 0.104   
## 4 D Premium 3513. 0.109   
## 5 D Ideal 2595. 0.217   
## 6 E Fair 3516. -0.172

ggplot(color\_cut, aes(color, price)) +  
 geom\_line(aes(group = cut), colour = "grey80") +  
 geom\_point(aes(colour = cut))



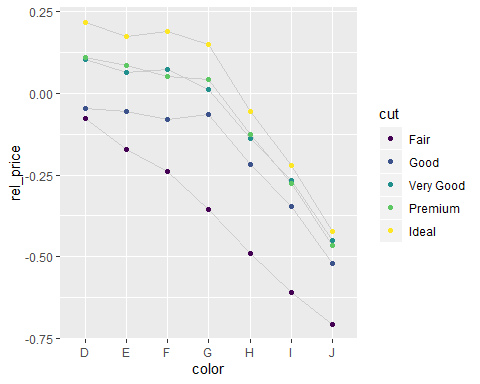
-> hard to ses how the quality of the diamond affects the price.

-> the lowest quality diamonds (fair cut with colour J) have the highest average value!

: those diamonds also tend to be larger. So size and quality are confounded.

- plot the relative price

ggplot(color\_cut, aes(color, rel\_price)) +  
geom\_line(aes(group = cut), colour = "grey80") +  
geom\_point(aes(colour = cut))



## 11.3 Texas Housing Data

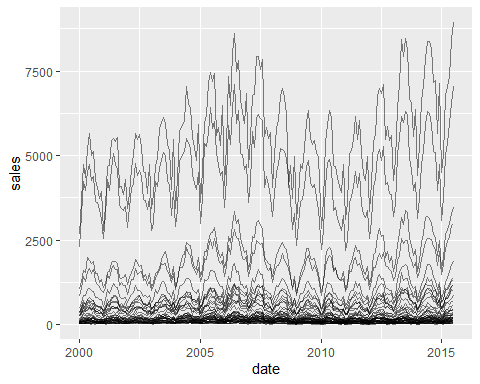
head(txhousing)

## # A tibble: 6 x 9  
## city year month sales volume median listings inventory date  
## <chr> <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Abilene 2000 1 72 5380000 71400 701 6.3 2000   
## 2 Abilene 2000 2 98 6505000 58700 746 6.6 2000.  
## 3 Abilene 2000 3 130 9285000 58100 784 6.8 2000.  
## 4 Abilene 2000 4 98 9730000 68600 785 6.9 2000.  
## 5 Abilene 2000 5 141 10590000 67300 794 6.8 2000.  
## 6 Abilene 2000 6 156 13910000 66900 780 6.6 2000.

- time series of sales for each city :

ggplot(txhousing, aes(date, sales)) +   
 geom\_line(aes(group = city), alpha = 0.5)

## Warning: Removed 430 rows containing missing values (geom\_path).



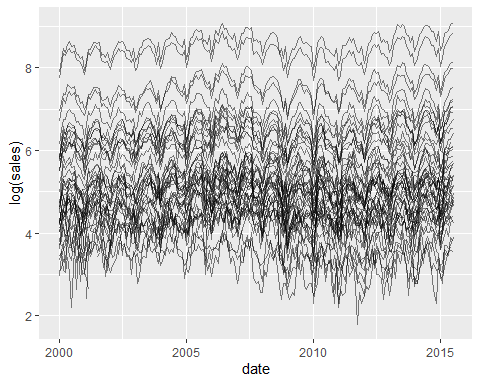
2 factors make it hard to see the long-term trend :

-> The range of sales varies

-> strong seasonal trend

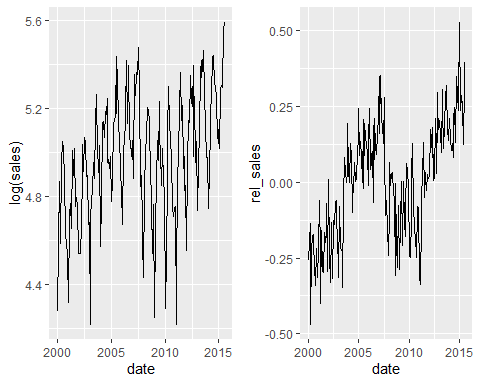
By log scales, can fix the first problem

ggplot(txhousing, aes(date, log(sales))) +  
 geom\_line(aes(group = city), alpha = 0.5).



Now, second problem!!

abilene <- txhousing %>% filter(city == "Abilene")  
  
g1 = ggplot(abilene, aes(date, log(sales))) +  
 geom\_line()  
  
mod <- lm(log(sales) ~ factor(month), data = abilene)  
  
abilene$rel\_sales <- resid(mod)  
  
g2 = ggplot(abilene, aes(date, rel\_sales)) +  
 geom\_line()  
  
grid.arrange(g1,g2,ncol=2)

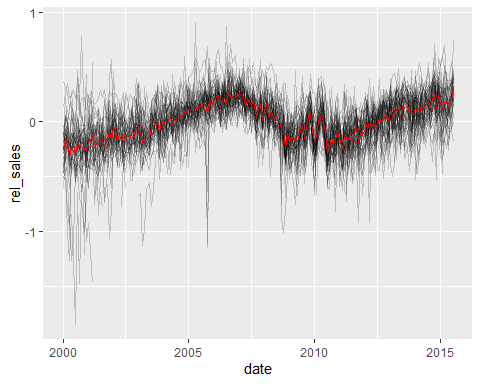


deseas <- function(x, month) {  
 resid(lm(x ~ factor(month), na.action = na.exclude))  
}  
  
txhousing <- txhousing %>%  
 group\_by(city) %>%  
 mutate(rel\_sales = deseas(log(sales), month))

ggplot(txhousing, aes(date, rel\_sales)) +  
 geom\_line(aes(group = city), alpha = 1/5) +  
 geom\_line(stat = "summary", fun.y = "mean", colour = "red")

## Warning: Removed 568 rows containing non-finite values (stat\_summary).

## Warning: Removed 430 rows containing missing values (geom\_path).



## 11.4 Visualising Models

*- do()* creates a new column called *mod*. : instead of containing an atomic vector, it’s a list. List can hold anything, including linear models!

models = txhousing %>%  
 group\_by(city) %>%  
 do(mod = lm(  
 log2(sales) ~ factor(month),  
 data = ., # . is a special pronoun in do(). It refers to the "current" data frame.  
 na.action = na.exclude  
 ))  
models

## Source: local data frame [46 x 2]  
## Groups: <by row>  
##   
## # A tibble: 46 x 2  
## city mod   
## \* <chr> <list>  
## 1 Abilene <lm>   
## 2 Amarillo <lm>   
## 3 Arlington <lm>   
## 4 Austin <lm>   
## 5 Bay Area <lm>   
## 6 Beaumont <lm>   
## 7 Brazoria County <lm>   
## 8 Brownsville <lm>   
## 9 Bryan-College Station <lm>   
## 10 Collin County <lm>   
## # … with 36 more rows

To visualise these models, We’ll do that with the **broom** package.

library(broom)

Broom provides 3 key verbs.

- *glance()* extracts **model**-level summaries with one row of data for each model.

- *tidy()* extracts **coefficient**-level summaries with one row of data for each coefficient.

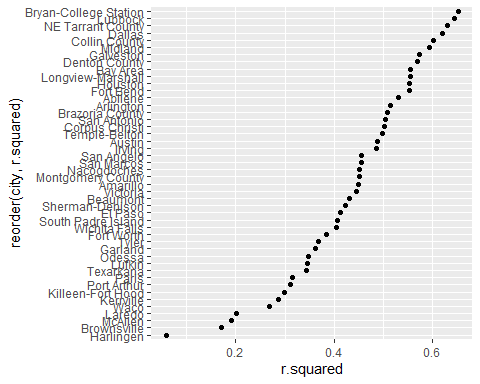
- *augment()* extracts **observation**-level summaries with one row of data for each observation in each model.

## 11.5 Model-Level Summaries

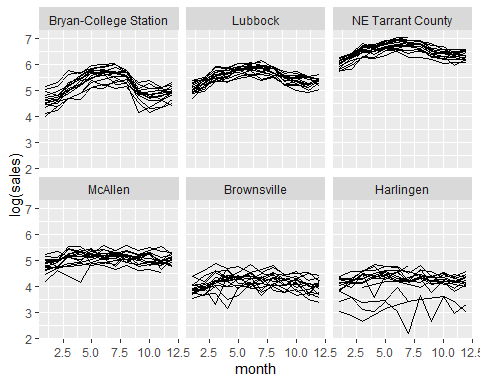
model\_sum <- models %>% glance(mod)  
model\_sum

## # A tibble: 46 x 12  
## # Groups: city [46]  
## city r.squared adj.r.squared sigma statistic p.value df logLik AIC  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <dbl> <dbl>  
## 1 Abil… 0.530 0.500 0.282 17.9 1.50e-23 12 -22.2 70.5  
## 2 Amar… 0.449 0.415 0.302 13.0 7.41e-18 12 -35.0 95.9  
## 3 Arli… 0.513 0.483 0.267 16.8 2.75e-22 12 -12.5 50.9  
## 4 Aust… 0.487 0.455 0.310 15.1 2.04e-20 12 -40.3 107.   
## 5 Bay … 0.555 0.527 0.265 19.9 1.45e-25 12 -10.5 47.0  
## 6 Beau… 0.430 0.395 0.275 12.0 1.18e-16 12 -18.0 62.1  
## 7 Braz… 0.508 0.475 0.292 15.1 6.48e-20 12 -26.3 78.5  
## 8 Brow… 0.171 0.119 0.420 3.25 4.59e- 4 12 -95.7 217.   
## 9 Brya… 0.651 0.629 0.406 29.7 1.73e-34 12 -90.6 207.   
## 10 Coll… 0.601 0.576 0.266 24.0 1.56e-29 12 -11.4 48.8  
## # … with 36 more rows, and 3 more variables: BIC <dbl>, deviance <dbl>,  
## # df.residual <int>

ggplot(model\_sum, aes(r.squared, reorder(city, r.squared))) +  
 geom\_point()



top3 <- c("Bryan-College Station", "Lubbock", "NE Tarrant County")  
bottom3 <- c("McAllen", "Brownsville", "Harlingen")  
  
extreme <- txhousing %>% ungroup() %>%  
 filter(city %in% c(top3, bottom3), !is.na(sales)) %>%  
 mutate(city = factor(city, c(top3, bottom3)))  
  
ggplot(extreme, aes(month, log(sales))) +  
 geom\_line(aes(group = year)) +  
 facet\_wrap(~city)



-> low have weaker seasonal patterns and more variation between years.

## 11.6 Coefficient-Level Summaries

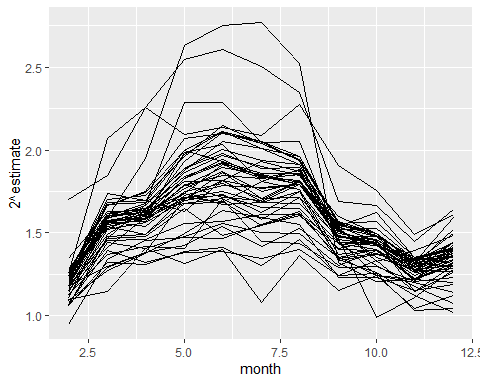
coefs = models %>% tidy(mod)  
coefs

## # A tibble: 552 x 6  
## # Groups: city [46]  
## city term estimate std.error statistic p.value  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Abilene (Intercept) 6.54 0.0704 92.9 7.90e-151  
## 2 Abilene factor(month)2 0.354 0.0996 3.55 4.91e- 4  
## 3 Abilene factor(month)3 0.675 0.0996 6.77 1.83e- 10  
## 4 Abilene factor(month)4 0.749 0.0996 7.52 2.76e- 12  
## 5 Abilene factor(month)5 0.916 0.0996 9.20 1.06e- 16  
## 6 Abilene factor(month)6 1.00 0.0996 10.1 4.37e- 19  
## 7 Abilene factor(month)7 0.954 0.0996 9.58 9.81e- 18  
## 8 Abilene factor(month)8 0.934 0.101 9.22 9.26e- 17  
## 9 Abilene factor(month)9 0.604 0.101 5.96 1.34e- 8  
## 10 Abilene factor(month)10 0.608 0.101 6.01 1.06e- 8  
## # … with 542 more rows

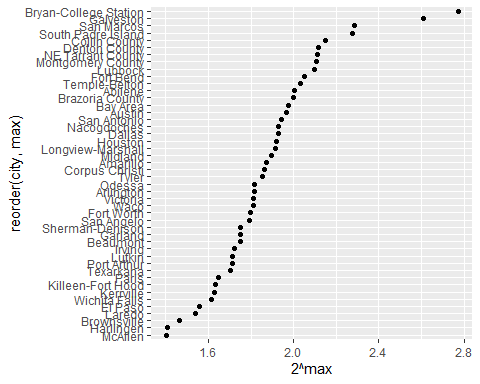
months <- coefs %>%  
 filter(grepl("factor", term)) %>%  
 tidyr::extract(term, "month", "(\\d+)", convert = TRUE)  
months

## # A tibble: 506 x 6  
## # Groups: city [46]  
## city month estimate std.error statistic p.value  
## <chr> <int> <dbl> <dbl> <dbl> <dbl>  
## 1 Abilene 2 0.354 0.0996 3.55 4.91e- 4  
## 2 Abilene 3 0.675 0.0996 6.77 1.83e-10  
## 3 Abilene 4 0.749 0.0996 7.52 2.76e-12  
## 4 Abilene 5 0.916 0.0996 9.20 1.06e-16  
## 5 Abilene 6 1.00 0.0996 10.1 4.37e-19  
## 6 Abilene 7 0.954 0.0996 9.58 9.81e-18  
## 7 Abilene 8 0.934 0.101 9.22 9.26e-17  
## 8 Abilene 9 0.604 0.101 5.96 1.34e- 8  
## 9 Abilene 10 0.608 0.101 6.01 1.06e- 8  
## 10 Abilene 11 0.419 0.101 4.14 5.45e- 5  
## # … with 496 more rows

ggplot(months, aes(month, 2 ^ estimate)) +  
 geom\_line(aes(group = city))



coef\_sum <- months %>%  
 group\_by(city) %>%  
 summarise(max = max(estimate))  
  
ggplot(coef\_sum, aes(2 ^ max, reorder(city, max))) +  
 geom\_point()



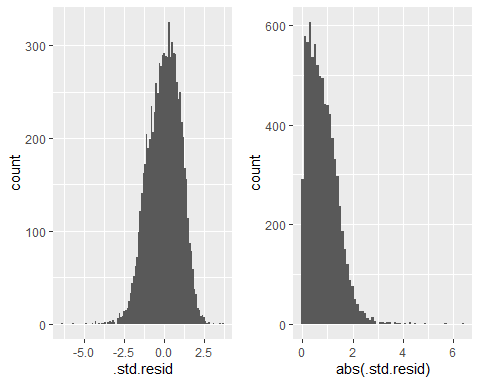
## 11.7 Observation Data

obs\_sum <- models %>% augment(mod)

obs\_sum

## # A tibble: 8,034 x 13  
## # Groups: city [46]  
## city log2.sales. factor.month. .fitted .se.fit .resid .hat .sigma .cooksd  
## <chr> <dbl> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Abil… 6.17 1 6.54 0.0704 -0.372 0.0625 0.281 0.0103   
## 2 Abil… 6.61 2 6.90 0.0704 -0.281 0.0625 0.282 0.00590  
## 3 Abil… 7.02 3 7.22 0.0704 -0.194 0.0625 0.282 0.00282  
## 4 Abil… 6.61 4 7.29 0.0704 -0.676 0.0625 0.278 0.0341   
## 5 Abil… 7.14 5 7.46 0.0704 -0.319 0.0625 0.281 0.00760  
## 6 Abil… 7.29 6 7.54 0.0704 -0.259 0.0625 0.282 0.00501  
## 7 Abil… 7.25 7 7.50 0.0704 -0.248 0.0625 0.282 0.00459  
## 8 Abil… 7.03 8 7.48 0.0727 -0.442 0.0667 0.280 0.0157   
## 9 Abil… 6.70 9 7.15 0.0727 -0.445 0.0667 0.280 0.0159   
## 10 Abil… 6.66 10 7.15 0.0727 -0.492 0.0667 0.280 0.0195   
## # … with 8,024 more rows, and 4 more variables: .std.resid <dbl>,  
## # .rownames <chr>, `log2(sales)` <dbl>, `factor(month)` <fct>

g1 = ggplot(obs\_sum, aes(.std.resid)) +  
 geom\_histogram(binwidth = 0.1)  
g2 = ggplot(obs\_sum, aes(abs(.std.resid))) +  
 geom\_histogram(binwidth = 0.1)  
  
grid.arrange(g1,g2,ncol=2)



obs\_sum %>%  
 filter(abs(.std.resid) > 2) %>%  
 group\_by(city) %>%  
 summarise(n = n(), avg = mean(abs(.std.resid))) %>%  
 arrange(desc(n))

## # A tibble: 43 x 3  
## city n avg  
## <chr> <int> <dbl>  
## 1 Texarkana 12 2.43  
## 2 Harlingen 11 2.73  
## 3 Waco 11 2.96  
## 4 Victoria 10 2.49  
## 5 Brazoria County 9 2.31  
## 6 Brownsville 9 2.48  
## 7 Bryan-College Station 9 2.27  
## 8 Killeen-Fort Hood 9 2.23  
## 9 Laredo 9 2.87  
## 10 Nacogdoches 9 2.15  
## # … with 33 more rows