

Lab: Data Visualisation

Actuarial Data Science Online Textbook

Fei Huang, UNSW Sydney

Learning Objectives

- Learn how to use tidyverse and ggplot() for data visualization.

Data Visualization

Data Source

While it might be difficult to obtain data to address a specific research problem or answer a business question, it is relatively easy to obtain data to test a model or an algorithm for data analysis. In the modern era, readers can obtain datasets from the Internet. The following is a list of some websites to obtain real-world data:

- **UCI Machine Learning Repository.** This website ([url: http://archive.ics.uci.edu/ml/index.php](http://archive.ics.uci.edu/ml/index.php)) maintains more than 400 datasets that can be used to test machine learning algorithms.
- **Kaggle.** The Kaggle website ([url: https://www.kaggle.com/](https://www.kaggle.com/)) include real-world datasets used for data science competitions. Readers can download data from Kaggle by registering an account.
- **DrivenData.** DrivenData aims at bringing cutting-edge practices in data science to solve some of the world's biggest social challenges. In its website ([url: https://www.drivendata.org/](https://www.drivendata.org/)), readers can participate in data science competitions and download datasets.
- **Analytics Vidhya.** This website ([url: https://datahack.analyticsvidhya.com/contest/all/](https://datahack.analyticsvidhya.com/contest/all/)) allows you to participate and download datasets from practice problems and hackathon problems.

- **KDD Cup.** KDD Cup is the annual Data Mining and Knowledge Discovery competition organized by the ACM Special Interest Group on Knowledge Discovery and Data Mining. This website (url: <http://www.kdd.org/kdd-cup>) contains the datasets used in past KDD Cup competitions since 1997.
- **U.S. Government's open data.** This website (url: <https://www.data.gov/>) contains about 200,000 datasets covering a wide range of areas including climate, education, energy, and finance.
- **AWS Public Datasets.** In this website (url: <https://aws.amazon.com/datasets/>), Amazon provides a centralized repository of public datasets, including some huge datasets.
- **CASdatasets: Insurance Datasets.** In this website (url: <http://cas.uqam.ca/>). A collection of datasets, originally for the book ‘Computational Actuarial Science with R’ edited by Arthur Charpentier. Now, the package contains a large variety of actuarial datasets.

Import Data

CASdatasets is R friendly, so we can download it by `install.packages("CASdatasets", repos = "http://cas.uqam.ca/pub/", type="source")`. After downloading, when you need this dataset, use `library()` function.

Before you install CASdatasets, make sure you have already installed these packages:

- `install.packages("zoo")`,
- `install.packages("xts")`,
- `install.packages("sp")`.

```
#Before you install CASdatasets, make sure you have already installed these packages:
#install.packages("zoo")
#install.packages("xts")
#install.packages("sp")
#install.packages("CASdatasets", repos = "http://cas.uqam.ca/pub/", type="source")

library(CASdatasets)
#?CASdatasets # Description of this dataset.

data(freMTPL2freq) # import a dataset we want
data(freMTPL2sev)

attach(freMTPL2freq) # The database is attached to the R search path. This means that the da
attach(freMTPL2sev)
freMTPL2freq$ClaimNb <- as.integer(freMTPL2freq$ClaimNb)
```

Now the datasets called `freMTPL2freq` and `freMTPL2sev` are imported successfully. Have a look at this dataset first. In the two datasets `freMTPL2freq`, `freMTPL2sev`, risk features are collected for 677,991 motor third-part liability policies (observed mostly on one year). In addition, we have claim numbers by policy as well as the corresponding claim amounts. `freMTPL2freq` contains the risk features and the claim number while `freMTPL2sev` contains the claim amount and the corresponding policy ID.

`freMTPL2freq` contains 12 columns:

- `IDpol`: The policy ID (used to link with the claims dataset).
- `ClaimNb`: Number of claims during the exposure period.
- `Exposure`: The period of exposure for a policy, in years.
- `Area`: The area code.
- `VehPower`: The power of the car (ordered categorical).
- `VehAge`: The vehicle age, in years.
- `DrivAge`: The driver age, in years (in France, people can drive a car at 18).
- `BonusMalus`: Bonus/malus, between 50 and 350: <100 means bonus, >100 means malus in France.
- `VehBrand`: The car brand (unknown categories).
- `VehGas`: The car gas, Diesel or regular.
- `Density`: The density of inhabitants (number of inhabitants per km²) in the city the driver of the car lives in.
- `Region`: The policy regions in France (based on a standard French classification).

`freMTPL2sev` contains 2 columns:

- `IDpol` The occurrence date (used to link with the contract dataset).
- `ClaimAmount` The cost of the claim, seen as at a recent date.

Task 1: How to know the relationship between claim frequency and driver age?

First, we create a figure using the codes from this week's lecture slide.

```
library(tidyverse)

ggplot(data = freMTPL2freq) + # the dataset
aes(x = DrivAge) + # the x position
aes(y = ClaimNb) +
aes(color = ClaimNb) +
# the y position
geom_point() + # the point geometric shape
# Adjust axis titles' font size
theme(axis.title=element_text(size=14,face="bold"))
```

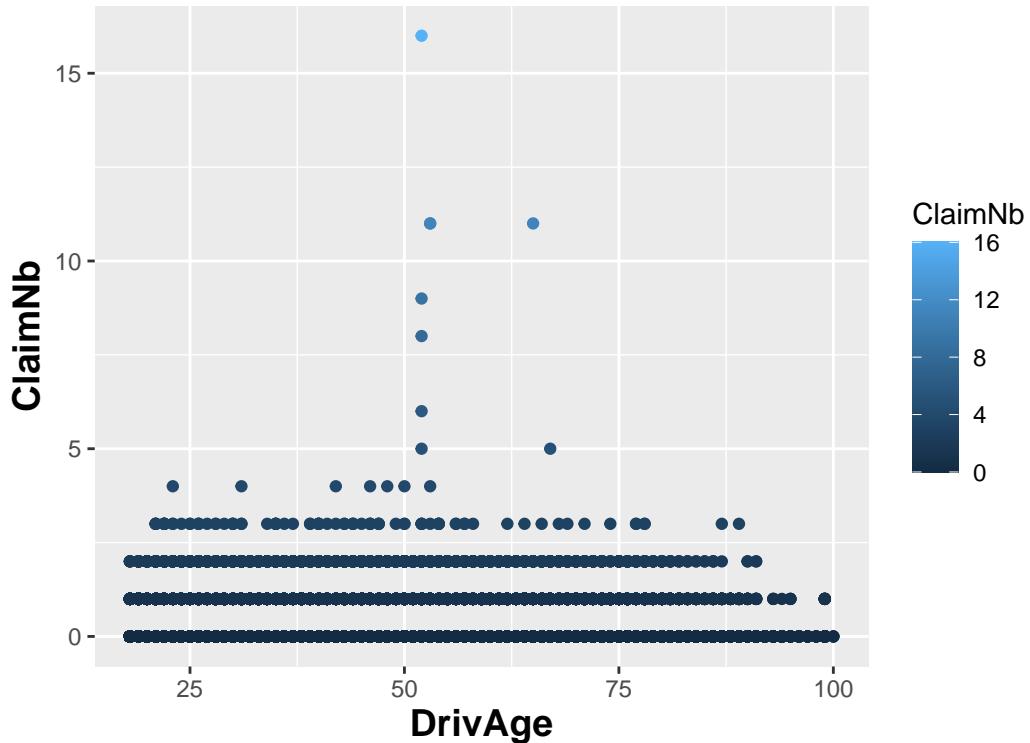


Figure 1: DriveAge v.s ClaimNb

It seems Figure @ref(fig:ageclaimnb) is not informative... **WHY?**

In practice, the frequency of most policy is 0. When you create plots in this way, you can see a bunch of data points at bottom level (0). And the number of observations is large (678013), which makes it difficult to recognize the pattern from so many points...

What I will do is plot the average `ClaimNb` for each `DriveAge`. This requires some code you haven't learned in lecture, but it is really helpful in this case.

```
freMTPL2freq %>%
  group_by(DrivAge) %>%
  summarize(AveClaimNb=mean(as.double(ClaimNb))) %>%
  ggplot(aes(x=DrivAge, y=AveClaimNb)) + geom_point() + geom_smooth()
```

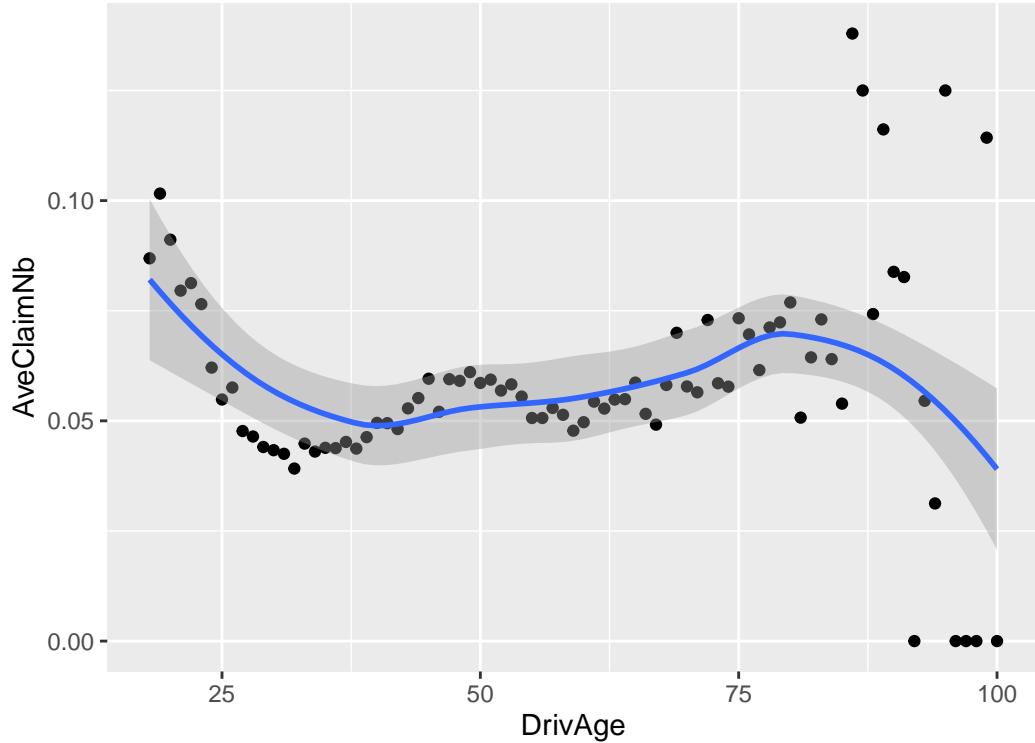


Figure 2: DriveAge v.s Average ClaimNB

From Figure @ref(fig:aveclaimnb), now we clearly see a pattern! It tells us younger ages less than 25 and older ages around 80 are more likely to make a claim. So, is this the true story?

In general insurance, it's common to calculate

$$\text{Claim Rate}_{age} = \frac{\sum_i \text{ClaimNb}_{age,i}}{\sum_i \text{Exposure}_{age,i}}$$

, where i represents the i th policyholder at this age . Claim Rate tells us the number of claim per exposure (year), which removes the effect of different exposures. Now let's see the relationship between Claim Rate and `DriveAge`.

```

freMTPL2freq %>%
  group_by(DrivAge) %>%
  summarize(claim_rate=sum(as.double(ClaimNb))/sum(Exposure)) %>%
  ggplot(aes(x=DrivAge, y=claim_rate)) + geom_point() + geom_smooth()

```

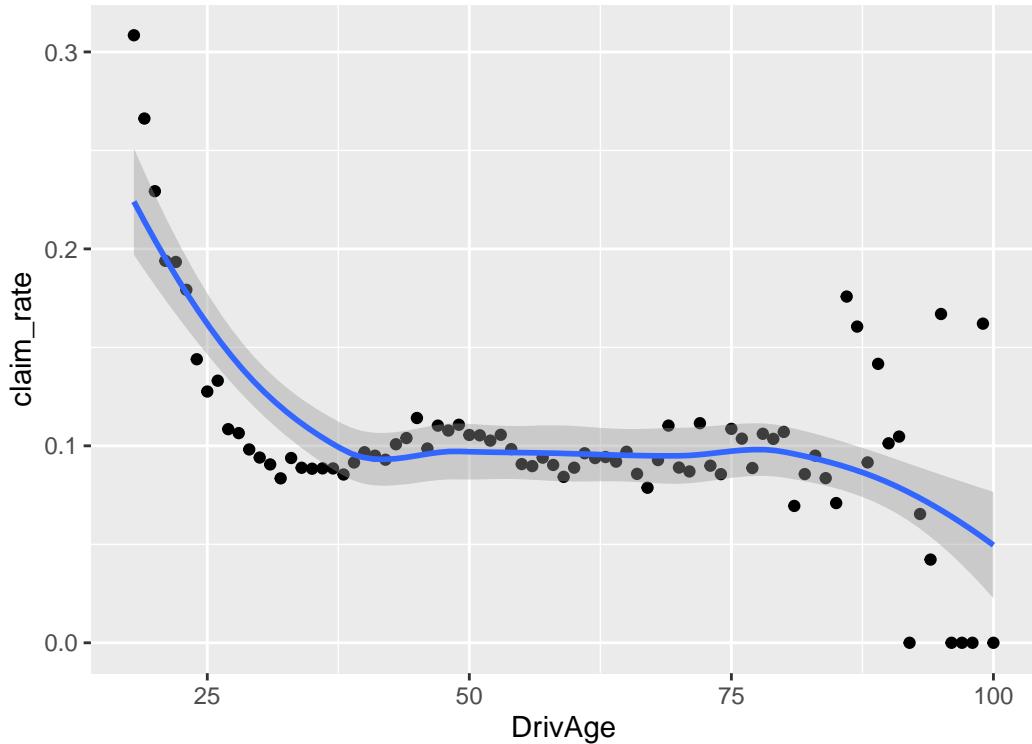


Figure 3: DriveAge v.s Average ClaimRate

From Figure @ref(fig:aveclaimrate), we can tell that younger ages become more dangerous while older ages are more safe.

Here is another informative figure called Violin plot. What do you learn from this plot?

```

# Violin plot DrivAge vs number of claims:
freMTPL2freq %>% filter(ClaimNb<5) %>%
  mutate(ClaimNb=as.factor(ClaimNb)) %>%
  ggplot(aes(ClaimNb, DrivAge)) +
  geom_violin(aes(fill = ClaimNb))

```

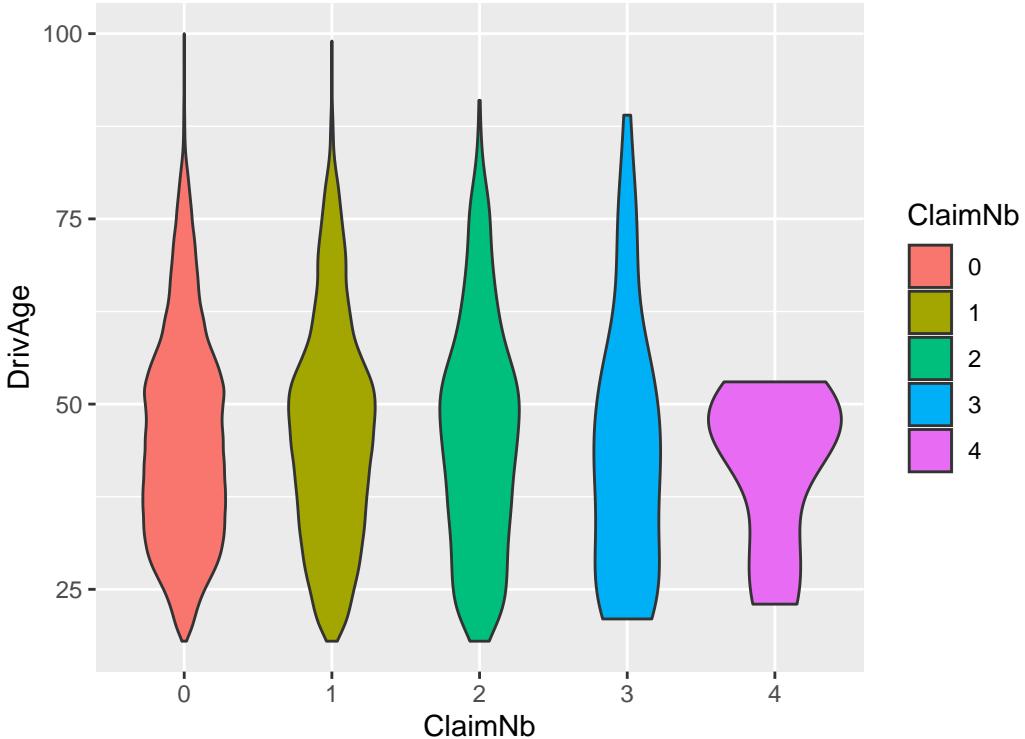


Figure 4: DriveAge v.s Average ClaimRate

Task 2: How to know the relationship between claim frequency and the area?

Task 2 is a little bit different with Task 1 because the type of Aera is factor and the type of DrivAge is integer.

```
str(freMTPL2freq)
```

```
'data.frame': 678013 obs. of 12 variables:
 $ IDpol     : num  1 3 5 10 11 13 15 17 18 21 ...
 $ ClaimNb   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Exposure   : num  0.1 0.77 0.75 0.09 0.84 0.52 0.45 0.27 0.71 0.15 ...
 $ VehPower   : int  5 5 6 7 7 6 6 7 7 7 ...
 $ VehAge     : int  0 0 2 0 0 2 2 0 0 0 ...
 $ DrivAge    : int  55 55 52 46 46 38 38 33 33 41 ...
 $ BonusMalus: int  50 50 50 50 50 50 50 68 68 50 ...
 $ VehBrand   : Factor w/ 11 levels "B1","B10","B11",...: 4 4 4 4 4 4 4 4 4 4 ...
 $ VehGas     : chr  "Regular" "Regular" "Diesel" "Diesel" ...
 $ Area       : Factor w/ 6 levels "A","B","C","D",...: 4 4 2 2 2 5 5 3 3 2 ...
 $ Density    : int  1217 1217 54 76 76 3003 3003 137 137 60 ...
```

```
$ Region      : Factor w/ 21 levels "Alsace","Aquitaine",...
```

First, we try the code in Task 1. Though we can get some information from Figure @ref(fig:crarea), it's not straightforward.

```
freMTPL2freq %>%
  group_by(Area) %>%
  summarise(claim_rate=sum(as.double(ClaimNb))/sum(Exposure)) %>%
  ggplot(aes(x=Area, y=claim_rate)) + geom_point() + geom_smooth() +
  coord_cartesian(ylim = c(0,0.2)) # set the axis limits
```

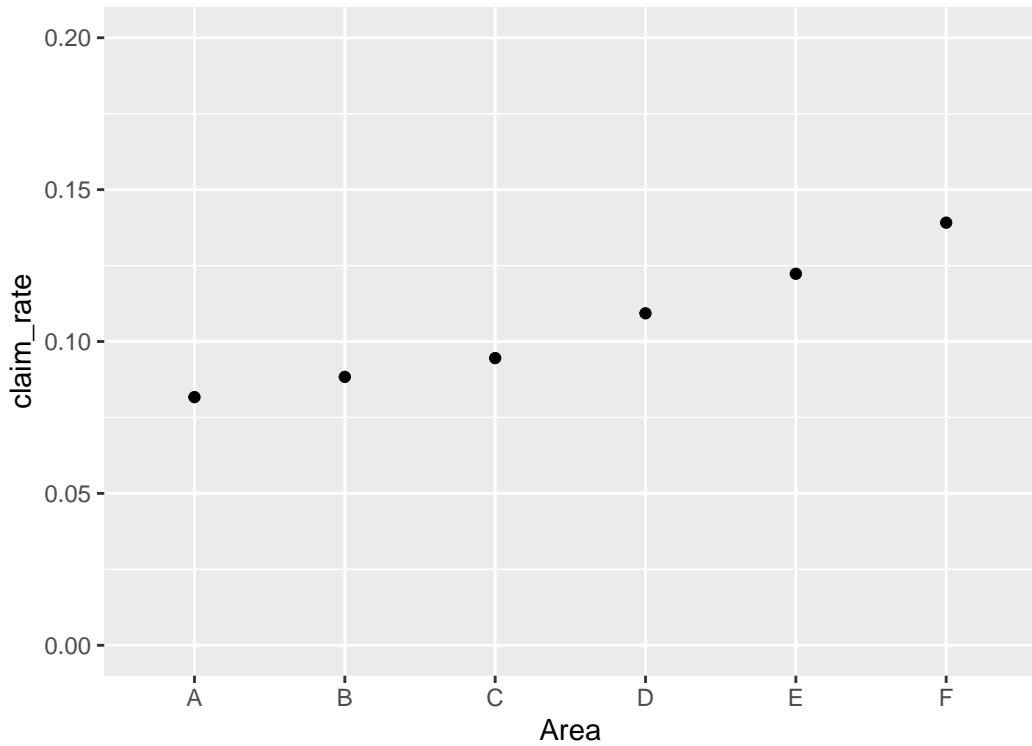


Figure 5: Area v.s Average ClaimRate

A better choice in this case is to use histogram (bar plot) for categorical variable. Now it's clear to compare the Claim Rates in different areas.

```
freMTPL2freq %>% # data piped into
group_by(Area) %>%
summarise(claim_rate=sum(as.double(ClaimNb))/sum(Exposure)) %>%
ggplot() + # initiating plot
aes(x = Area) + #categorical variable
```

```

aes(y = claim_rate) +
geom_col() + #Use `geom_col` to create a column geometry
aes(color = Area) +
aes(fill = Area) + # new aes 'fill'
aes(linetype = Area)+ #new aes 'linetype'
coord_cartesian(ylim = c(0,0.2)) # set the axis limits

```

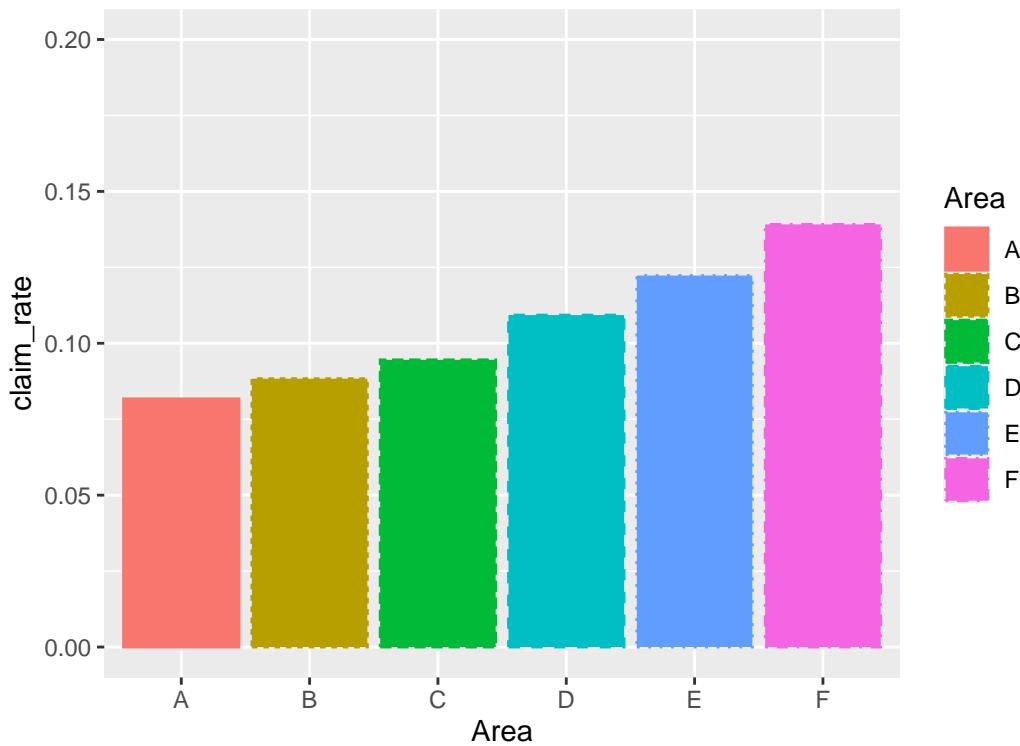


Figure 6: Area v.s Average ClaimRate (histogram)