ACSC/STAT 3740, Predictive Analytics

WINTER 2023

Toby Kenney

Homework Sheet 1

Model Solutions

[Note: all data in this homework are simulated.]

[Note: With many of these problems, there is no "correct" solution. These model solutions give a range of reasonable approaches, but there are many other good approaches that could be taken.]

Common `ggplot` settings

These settings are used for a number of plots in this homework. For clarity, we define them as variables at the start of our script.

```
library(dplyr)
### For various commands to tidy up the data mostly using the %>% operator.

library(ggplot2)
### For various plotting commands.

library(scales)
### For various commands that adjust scales, such as trans_new, which
### creates a new axis transformation.


### This defines a theme that can be added to any ggplot to make the text
### larger. It is used in many of the solutions for this homework.

largertextsize<-theme(plot.title=element_text(size=20,hjust=0.5),
                      axis.title=element_text(size=20,hjust=0.5),
                      axis.text=element_text(size=16),
                      legend.title=element_text(size=20,hjust=0.5),
                      legend.text=element_text(size=16),
                      strip.text=element_text(size=16))

### Not all of these are used for every plot.
### Most of the names are obvious. strip.text is for titles of
### subplots made with facet_wrap or facet_grid.
```

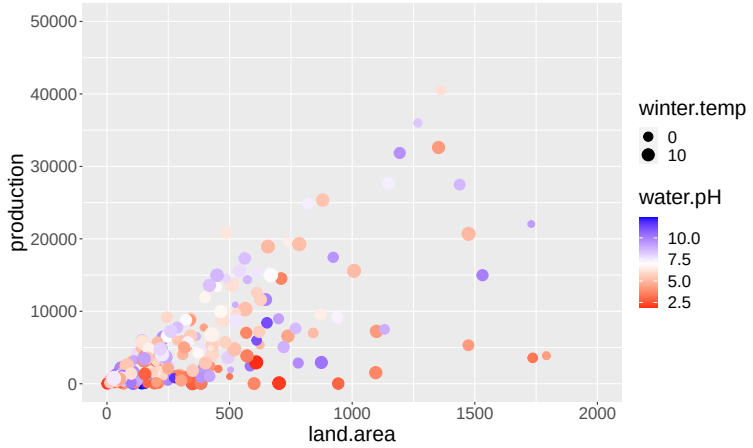
Basic Questions

1. *The file* `HW1Q1.txt` *is from an experiment about the effect of acid rain on forestry. It includes the following variables:*

Variable name	*Meaning*
land.area	*The area of the plot of land in square km*
yearly.rain	*The total annual rainfall in mm*
soil.type	*The predominant type of soil*
winter.temp	*The average daytime maximum temperature during winter months*
summer.temp	*The average daytime maximum temperature during summer months*
water.pH	*The pH of the groundwater in the region*
production	*Annual lumber production in tonnes*

Display this data set in a plot.

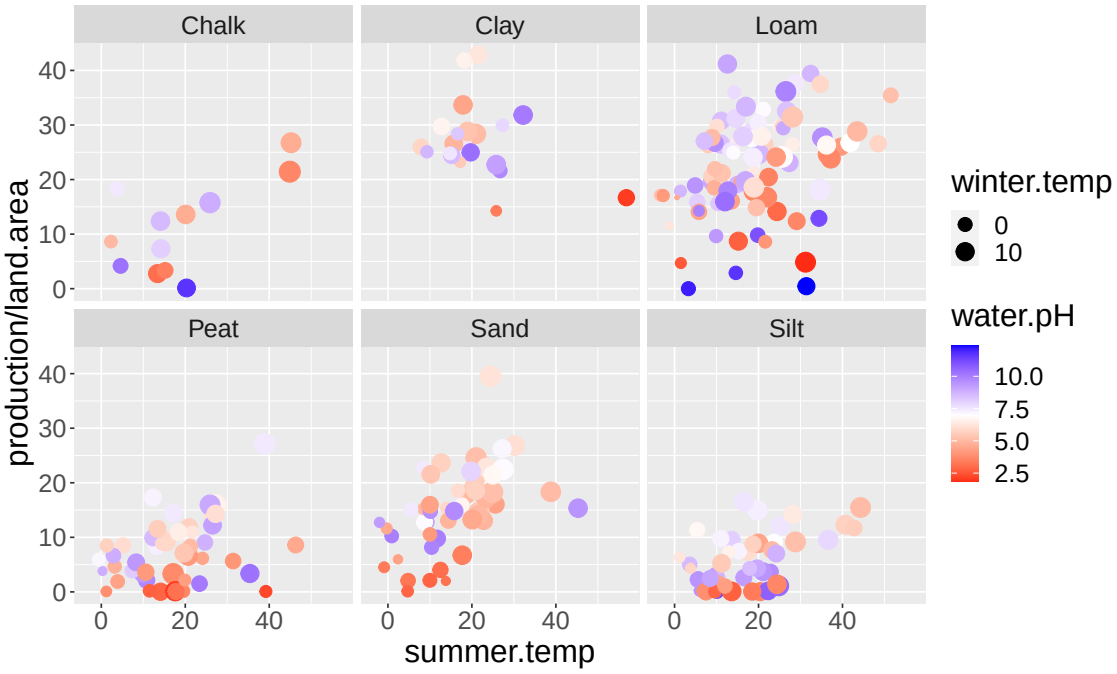
As the response variable, production would usually be indicated on the y-axis. Because of the large outlier, we should either log-transform this value, or else limit the range to remove some outliers. pH can be represented using colour. Since pH is a two-sided scale, centred at 7, it makes sense to use a 2-sided colour scale. It is traditional to use red for acid (low pH) and blue for alkali (high pH). We can also use size to represent `summer.temp` or `winter.temp`.



This was produced with the code

```
ggplot(HW1Q1,
       mapping=aes(y=production,
                   x=land.area,
                   colour=water.pH,
                   size=winter.temp))+
    geom_point()+
    scale_colour_gradient2(midpoint=7,low="red",high="blue")+
    largertextsize+
    scale_x_continuous(limits=c(0,2000))+
    scale_y_continuous(limits=c(0,50000))
```

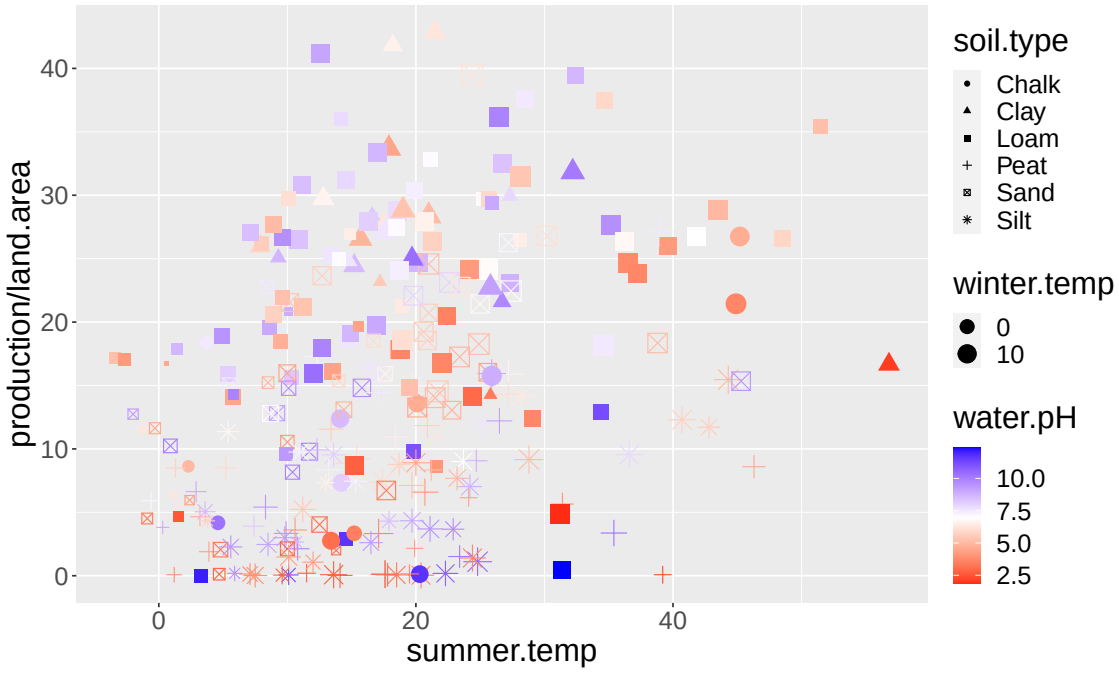
Given the obvious relation between `land.area` and `production`, another approach is to plot the ratio $\frac{\texttt{production}}{\texttt{land.area}}$. This allows us to represent, for example, `summer.temp` with x-coordinate and `winter.temp` with size. Since there are a relatively small number of soil types, a `facet_wrap` might be an appropriate way to show them.



This was produced with the code

```
ggplot(HW1Q1,
       mapping=aes(y=production/land.area,
                   x=summer.temp,
                   colour=water.pH,
                   size=winter.temp))+
    geom_point()+
    facet_wrap(soil.type~.)+
    scale_colour_gradient2(midpoint=7,low="red",high="blue")+
    largertextsize
```

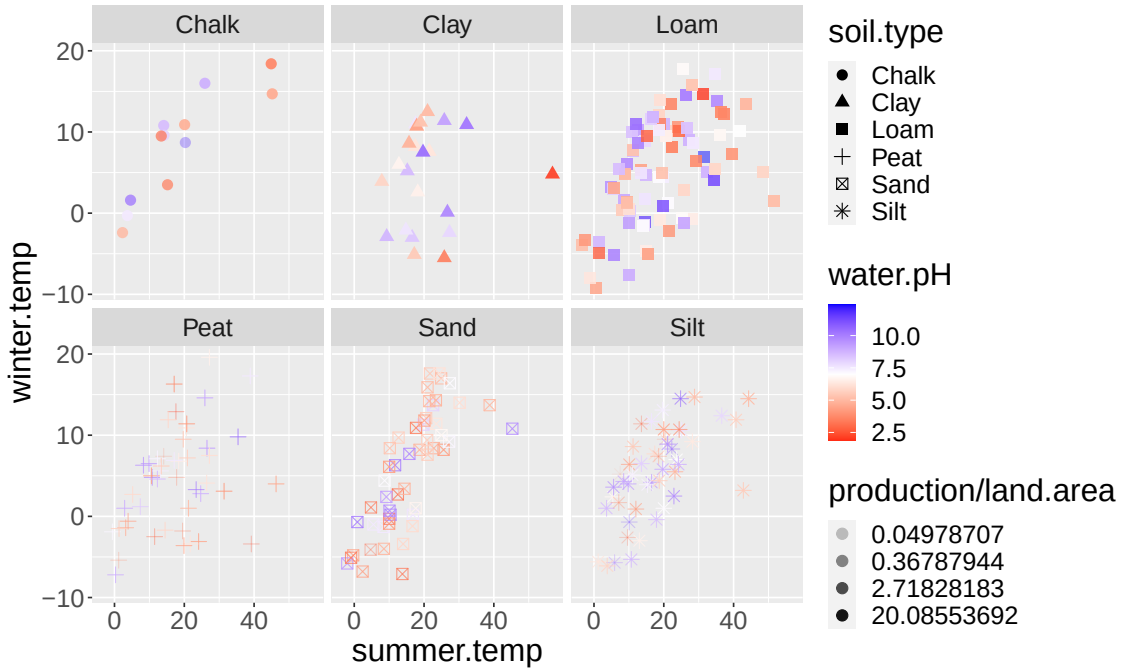
Using shape for `soil.type` is also possible, as in the following plot:




```
ggplot(HW1Q1,
       mapping=aes(y=production/land.area,
                   x=summer.temp,
                   colour=water.pH,
                   size=winter.temp,
                   shape=soil.type))+
    geom_point()+
    scale_colour_gradient2(midpoint=7,low="red",high="blue")+
    largertextsize
```

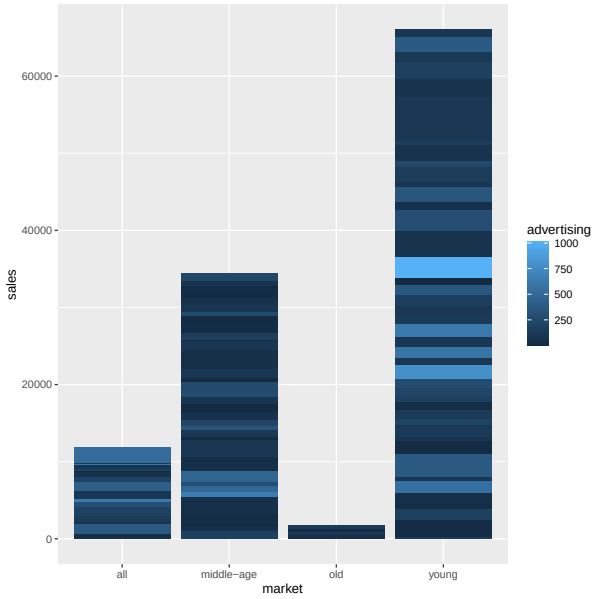
However, in this case, I find it harder to see the effect of soil type in this plot than in the `facet_wrap`. In some cases, when the variance is smaller for each soil type, using shape can be easier to see, so the choice between shape and facet wrap needs to be carefully considered.

Another possibility is to use y-coordinate to represent winter temperature, and use another channel to represent production per unit land area. Alpha (transparancy) works reasonably well, using a log-transformation so that the variation on each facet is noticable.




```
ggplot(HW1Q1, mapping=aes(alpha=production/land.area,
                          x=summer.temp,
                          y=winter.temp,
                          shape=soil.type,
                          colour=water.pH))+
    geom_point(size=3)+
    largertextsize+
    facet_wrap(soil.type~.)+
    scale_colour_gradient2(midpoint=7,low="red",high="blue")+
    scale_alpha_continuous(trans="log")
```

2. *A retail company is studying the effect of advertising on sales, and has produced the plot below. Identify at least three issues with the plot and produce a new plot that better displays the data.*



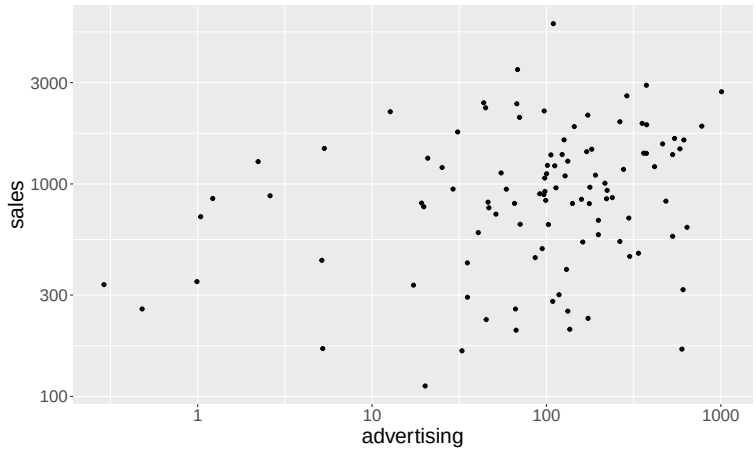
Provide `R` *code for the new plot. [The data used to produce the figure is in the file* `HW1Q2.txt`*. You should include more information from that file in the plot as appropriate.]*

(i) The stacked bar-chart makes it very difficult to see the individual products. The total sales for each market do not seem to be very relevant to the question under consideration.

(ii) The graph puts more emphasis on the large-sales items, and pays less attention to the more numerous items with limited sales.

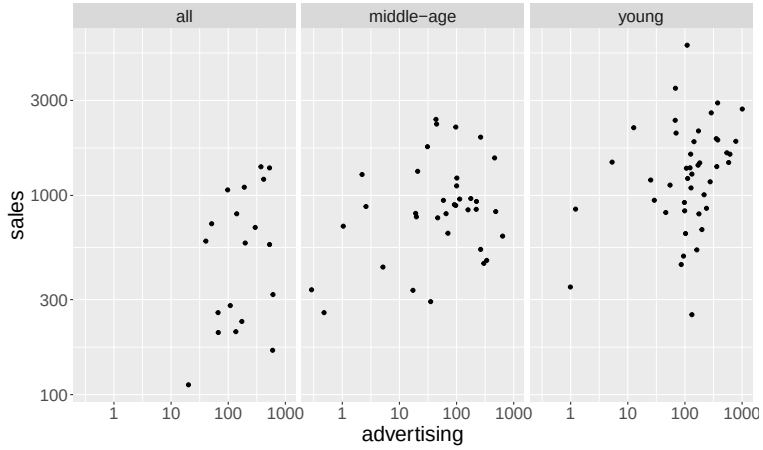
(iii) The graph does not show price or brand, which are highly related to sales.

A first attempt might be a scatterplot of advertising against sales. Because of the skewed distribution of both advertising and sales, it makes sense to log-transform both axes.



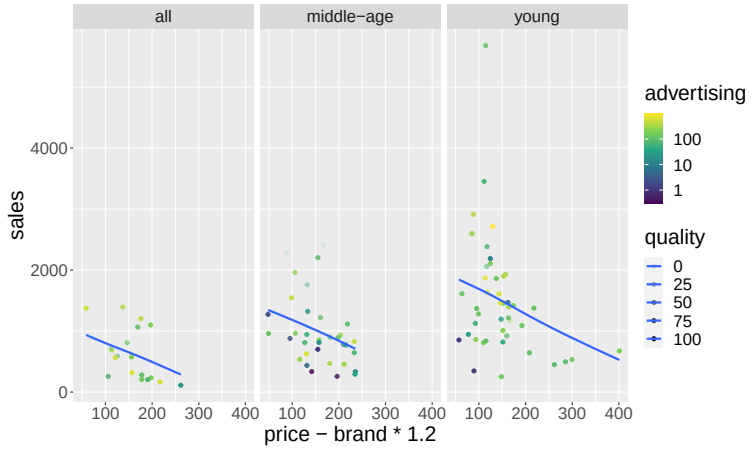

```
ggplot(HW1Q2,
       mapping=aes(x=advertising,
                   y=sales))+
    geom_point()+
    largertextsize+
    scale_x_log10()+
    scale_y_log10()
```

This plot does not show a strong correlation because it ignores the other predictors. To make a better plot, we should include price, market and brand, and ideally also quality, though the skewed distribution of quality makes this less crucial. Given the small number of classes for market, a `facet_wrap` is appropriate. Since there are very few products for the `old` market, it makes sense to remove that.



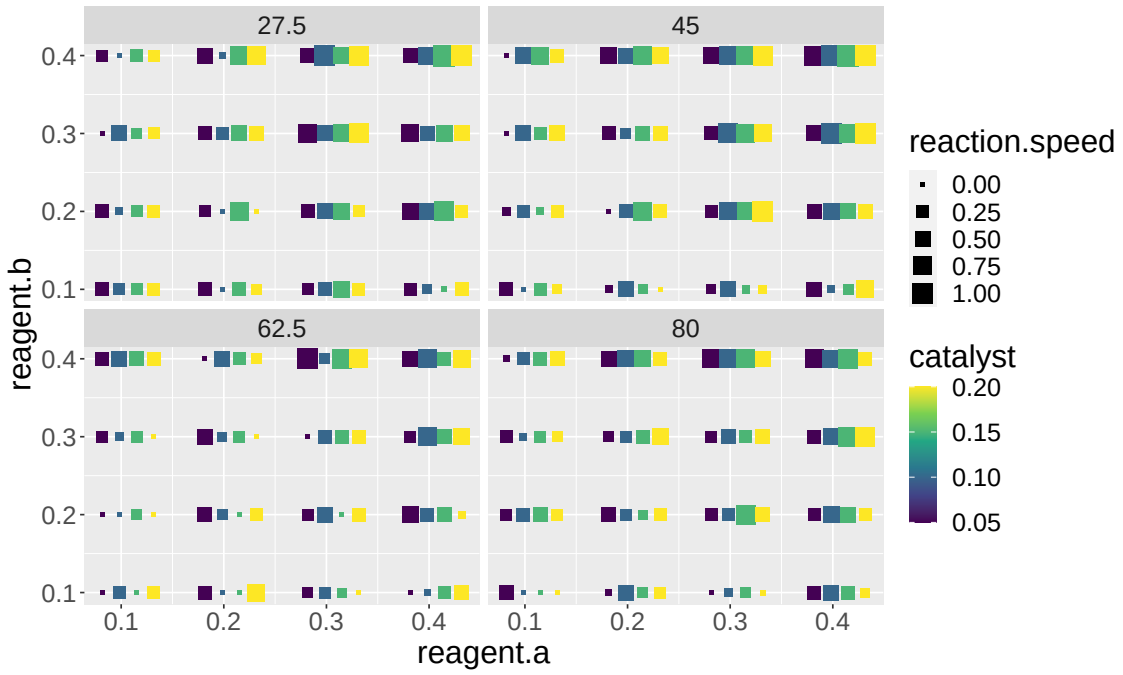
```
library(dplyr)

ggplot(HW1Q2%>%filter(market!="old"),
       mapping=aes(x=advertising,
                   y=sales))+
    geom_point()+
    largertextsize+
    scale_x_log10()+
    scale_y_log10()+
    facet_wrap(market~.)
```

We still need to include the other predictors. One approach to including so many predictors is to include a combination of the predictors. After some experimentation, `price` $-$ $1.2 *$ `brand` seems a reasonable combination to show. We can also add a trend line to make it easier to see patterns.




```
ggplot(HW1Q2%>%filter(market!="old"),
       mapping=aes(y=sales,
                   x=price-brand*1.2,
                   colour=advertising,
                   alpha=quality))+
    geom_point()+
    largertextsize+
    facet_wrap(market~.)+
    scale_colour_viridis_c(trans="log",breaks=c(1,10,100))+
    geom_smooth(method="gam",se=FALSE)
```

3. *Use* `ggplot` *to produce the following plot from the data in file* `HW1Q3.txt`*. [Make sure to reproduce all aspects of the plot — axis scales, labels, etc.]*



The code that originally produced the figure is

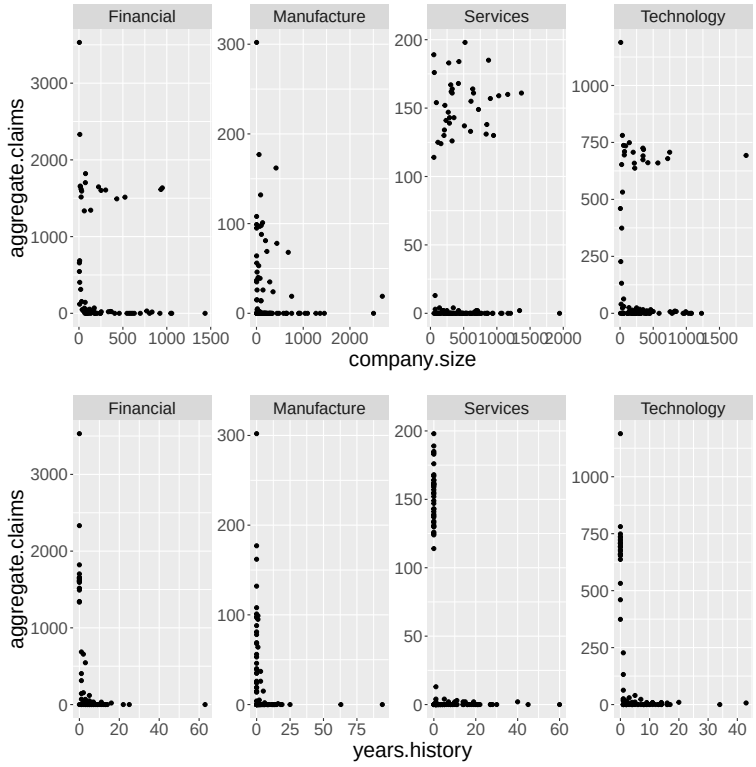
```
ggplot(HW1Q3,
       mapping=aes(x=reagent.a+catalyst/3-0.035,
                   y=reagent.b,
                   colour=catalyst,
                   size=reaction.speed))+
    geom_point(shape=15)+
    facet_wrap(temp~.)+
    scale_colour_viridis_c()+
    largertextsize+
    scale_x_continuous(name="reagent.a")
```

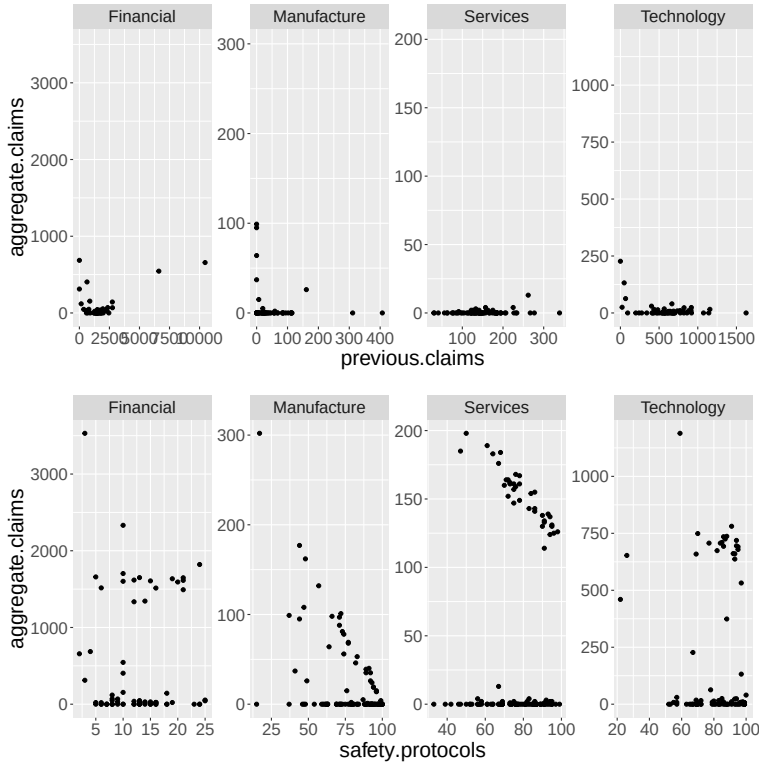
4. *The file* `HW1Q4.txt` *contains the following data from an insurance company about worker's compensation insurance claims.*

Variable	*Meaning*
company.size	*The number of workers at the company*
industry	*The industry of the company*
years.history	*The number of years of history with this company*
previous.claims	*The per-worker average previous annual claims from the company*
safety.protocols	*An index indicating how many safety protocols the company adheres to*
aggregate.claims	*The company's aggregate average claims per worker.*

Construct a plot or plots to show these data for the purpose of data exploration.

There are a number of plots we might make. It is natural to start with pairwise scatterplots of individual predictors and the response. Because of the differences between different industries, a `facet_wrap` is appropriate.





We use free scales for the facets because the different industries have very different properties, and we are more interested in the relations between variables within each industry, rather than the differences between industries. The code to produce these plots is:

```
ggplot(HW1Q4,
       mapping=aes(x=company.size,
                   y=aggregate.claims))+
    geom_point()+
    facet_wrap(industry~.,scales="free",nrow=1)+
    largertextsize
```

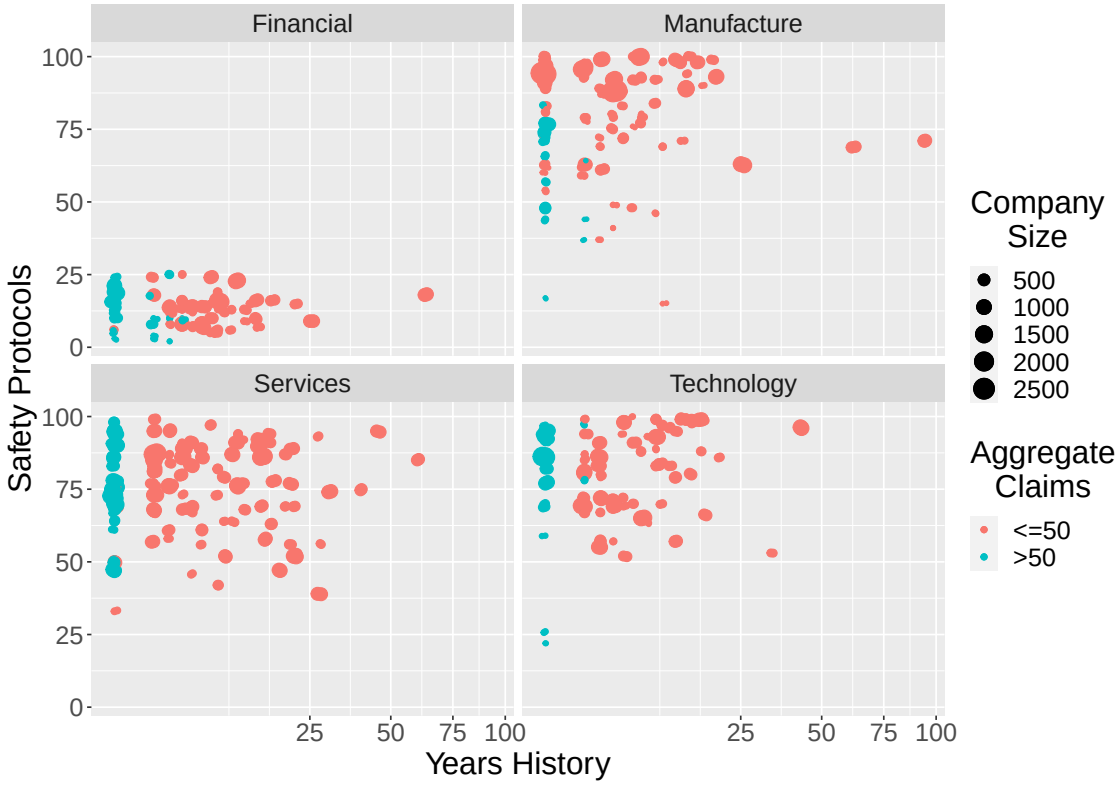
```
ggplot(HW1Q4,mapping=aes(x=years.history,
                         y=aggregate.claims))+
    geom_point()+
    facet_wrap(industry~.,scales="free",nrow=1)+
    largertextsize
```

```
ggplot(HW1Q4, mapping=aes(x=previous.claims,
                          y=aggregate.claims))+
    geom_point()+
    facet_wrap(industry~., scales="free", nrow=1)+
    largertextsize
```

```
ggplot(HW1Q4, mapping=aes(x=safety.protocols,
                          y=aggregate.claims))+
    geom_point()+
    facet_wrap(industry~., scales="free", nrow=1)+
    largertextsize
```

One of the immediately obvious features of these plots is the bimodal distribution of the aggregate claims, with many aggregate claims close to zero, and others having large values. Many of these larger values are not shown on the previous claims plot because those values are NA (due to no previous history).

Sometimes a log transformation can give a better distribution for the response variable. However, in this case, there seems to be a linear relation between `safety.protocols` and the larger `aggregate.claims` values. Therefore, a better approach might be to separate the data into small aggregate claims, and large aggregate claims. We can first show which companies have small aggregate claims, and which have large aggregate claims:

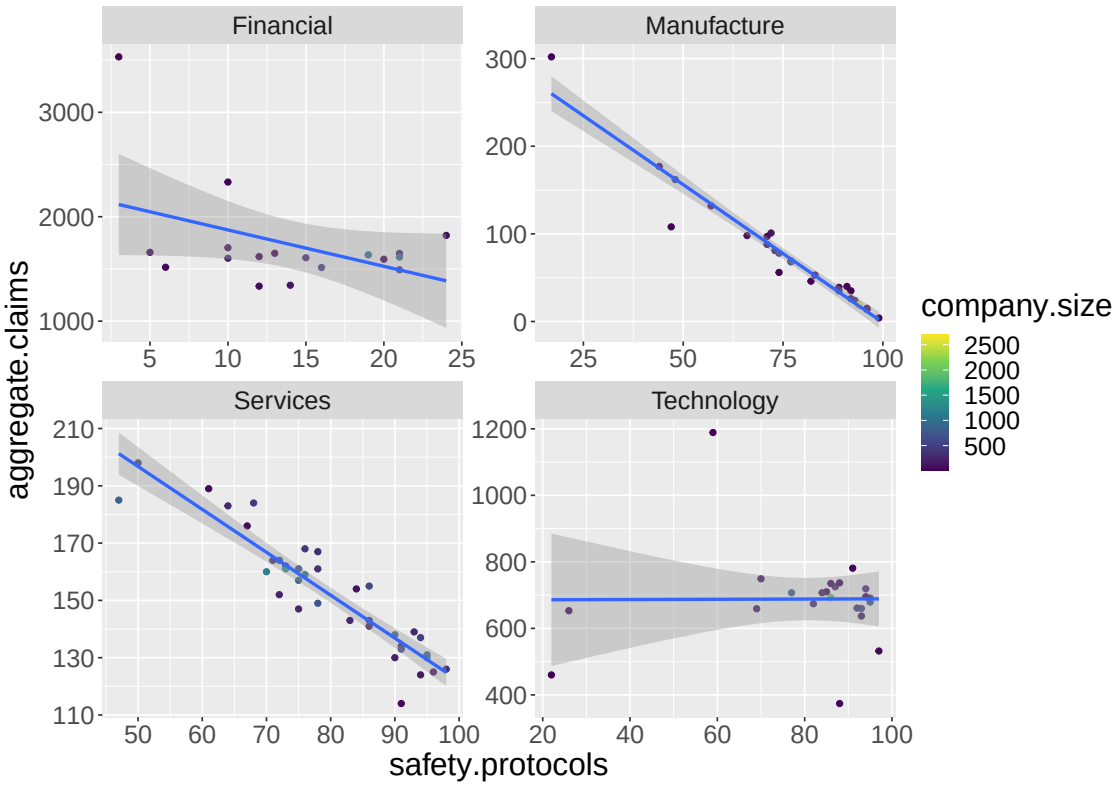


I have arbitrarily used 50 as the cut-off between small and large, but other similar values would work reasonably well. Because of the large number of companiess with 0 years previous history, I have added a `geom_jitter` to help see how many points at a given position. [An alternative approach would be to set partial transparency so that overlapping points result in deeper colours.] I have used a square root transformation on years.history, to partially correct the skewed distribution. A log transformation would give $-\infty$ values, which could be problematic, particularly with the `geom_jitter`.

This plot is produced using the code.

```
ggplot(HW1Q4, mapping=aes(size=company.size,
                          x=years.history,
                          colour=aggregate.claims>50,
                          y=safety.protocols))+
    geom_point()+
    largertextsize+
    scale_x_continuous(trans="sqrt",name="Years History")+
    scale_y_continuous(name="Safety Protocols")+
    scale_colour_discrete(name="Aggregate\nClaims",labels=c("<=50",">50"))+
    scale_size_continuous(name="Company\nSize")+
    geom_jitter(width=0.1)+
    facet_wrap(industry~.)
```

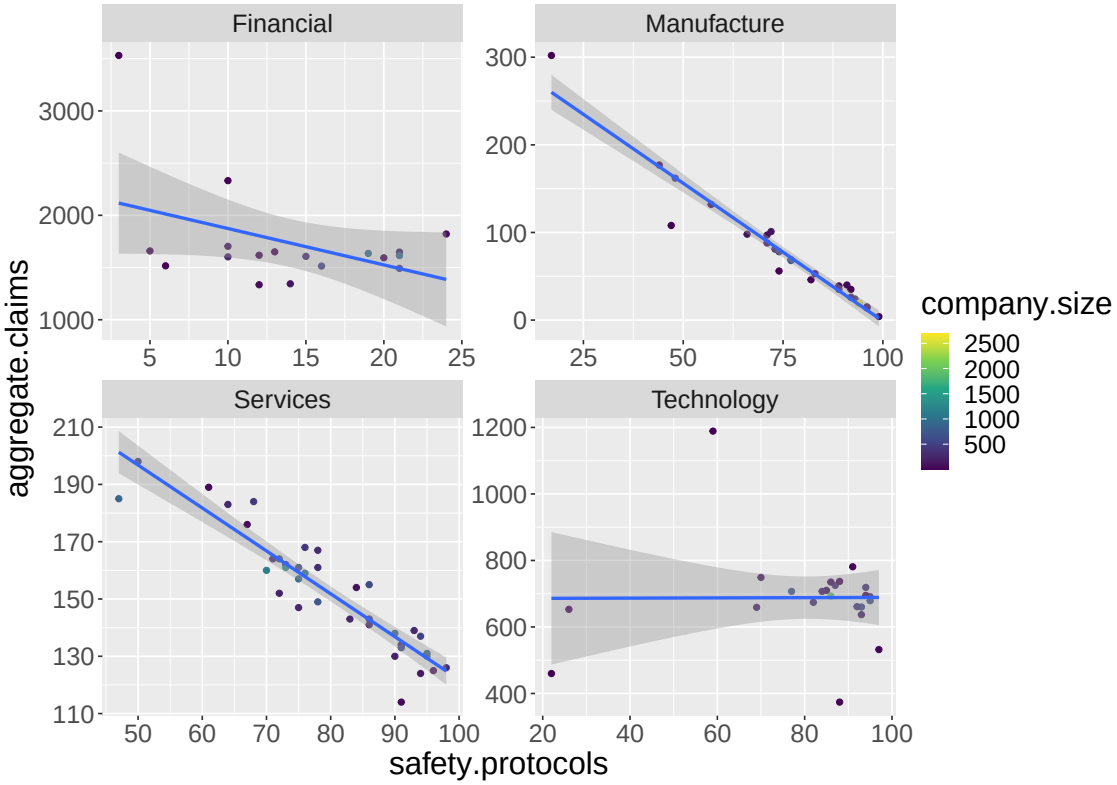
We may then plot just the points with 0 years history and aggregate claims exceeding 1.

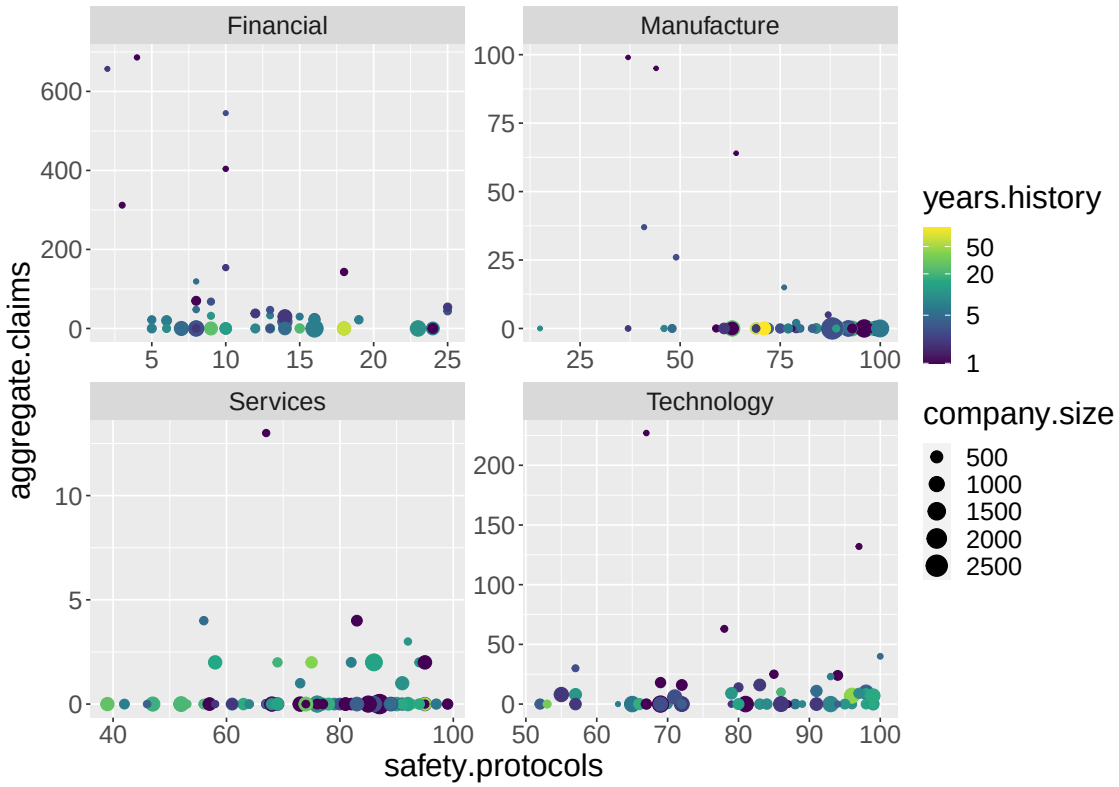


I have added trend lines to this plot to both highlight the clear trends, and also to allow easier comparison of points with the trend. This is produced by the code:

```
ggplot(HW1Q4%>%filter(years.history==0&
                      aggregate.claims>1),
       mapping=aes(y=aggregate.claims,
                   x=safety.protocols,
                   colour=company.size))+
    geom_point()+
    largertextsize+
    facet_wrap(industry~.,scale="free")+
    scale_colour_viridis_c()+
    geom_smooth(method="lm")
```

Finally, it makes sense to plot the points with more than 0 years previous history in another plot or plots. I chose to make two plots for these — one with `safety.protocols` on the x axis, and one with `previous.claims` on the x axis. Perhaps all the predictors could be shown on a single plot, but this would be likely to result in using a difficult channel for at least one of them.





The code to produce these plots is:

```
ggplot(HW1Q4%>%filter(years.history>0),
       mapping=aes(y=aggregate.claims,
                   x=safety.protocols,
                   colour=years.history,
                   size=company.size))+
    geom_point()+
    largertextsize+
    facet_wrap(industry~.,scale="free")+
    scale_colour_viridis_c(trans="log",breaks=c(1,5,20,50))
```

```
ggplot(HW1Q4%>%filter(years.history>0),
       mapping=aes(y=aggregate.claims,
                   x=previous.claims,
                   colour=years.history,
                   size=company.size))+
    geom_point()+
    largertextsize+
    facet_wrap(industry~.,scale="free")+
    scale_colour_viridis_c(trans="log",breaks=c(1,5,20,50))
```

Standard Questions

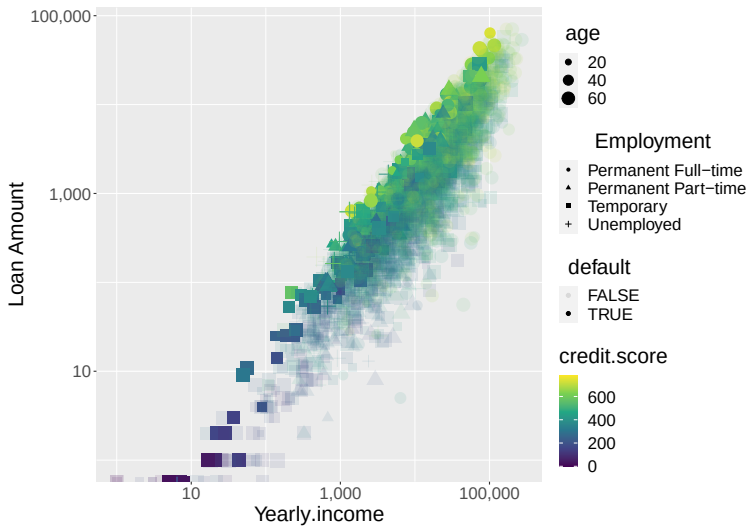
5. *A bank collects the following data on loan repayments by customers. The data are contained in the file* `HW1Q5.txt` *and include the following variables:*

Variable	*Meaning*
loan.amount	*The size of the loan*
yearly.income	*The borrower's annual income*
credit.score	*The borrower's credit score*
age	*The borrower's age*
employment.status	*The borrower's employment status*
default	*Whether the loan is repaid*

Make a plot to show these data.

There are four continuous variables and two discrete variables, including the response. The first two continuous variables `loan.amount` and `yearly.income` are both skewed, so a log transformation might be appropriate. An alternative possible transformation is the ratio $\frac{\texttt{loan.amount}}{\texttt{yearly.income}}$.

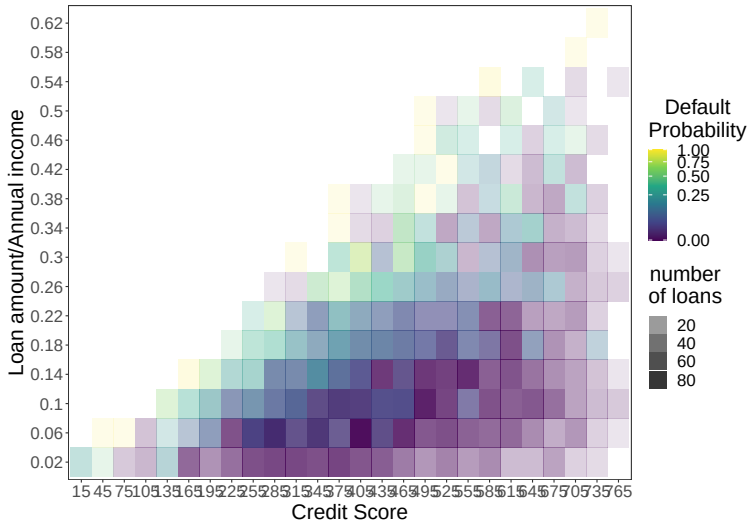
Another important feature is that defaults are rare. It is possible to plot all the information on a single plot, for example



using code

```
ggplot(HW1Q5, mapping=aes(y=loan.amount,
                          x=yearly.income,
                          colour=credit.score,
                          size=age,
                          shape=employment.status,
                          alpha=default))+
    geom_point()+
    scale_x_log10(name="Yearly.income",labels=scales::comma)+
    scale_y_log10(name="Loan  Amount",labels=scales::comma)+
    scale_shape_discrete(name="Employment")+
    scale_colour_viridis_c()+
    largertextsize
```

However, it is difficult to read a lot of the patterns from this plot. One problem is that it is hard to see how many non-defaulting loans are at a given location on the plot. We can use a tile plot to show the probability of default for each location:

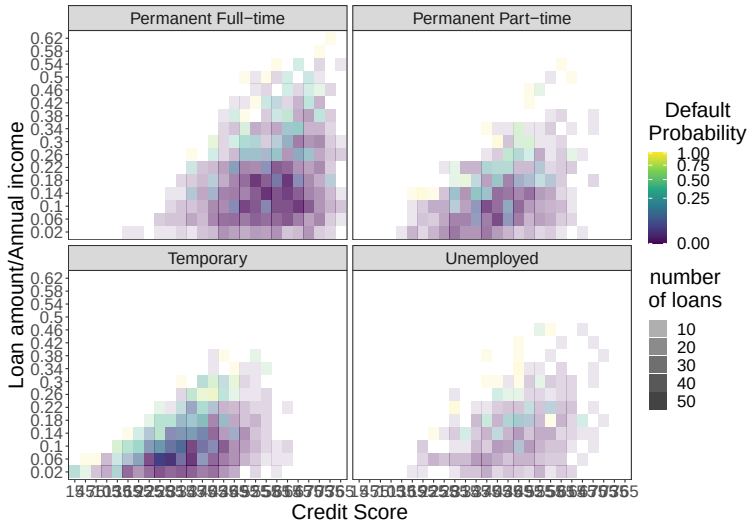


using code

```
ggplot(HW1Q5%>%mutate(lr=cut(loan.amount/yearly.income,breaks=(0:25)/25-1e-20),
                      cb=cut(credit.score,breaks=(0:30)*30-1e-20))%>%
       group_by(lr,cb)%>%summarise(total=n(),def.prob=mean(default)),
       mapping=aes(y=lr,x=cb,alpha=total,fill=def.prob))+
    geom_tile()+
    largertextsize+
    scale_alpha_continuous(name="number\nof loans",trans="sqrt")+
    scale_fill_viridis_c(name="Default\nProbability",trans="sqrt")+
    scale_x_discrete(name="Credit Score",labels=(0:30)*30+15)+
    scale_y_discrete(name="Loan amount/Annual income",labels=(0:25)/25+0.02)+
    theme_bw()+
    theme(panel.grid.major=element_blank())+
    largertextsize
```

I have removed the guidelines from this plot as they are distracting. I have used transparency to indicate the number of loans in each tile, so that tiles based on many observations (with probabilities that are therefore less reliable) are fainter. I have used the ratio $\frac{\texttt{loan.amount}}{\texttt{yearly.income}}$ to reduce the number of predictors that need to be plotted on the graph.

Unfortunately, this graph does not show the effect of age or employment status. It is not too challenging to add a `facet_wrap` on employment status. Note that we need to group by employment status in order for the `facet_wrap` to be possible.



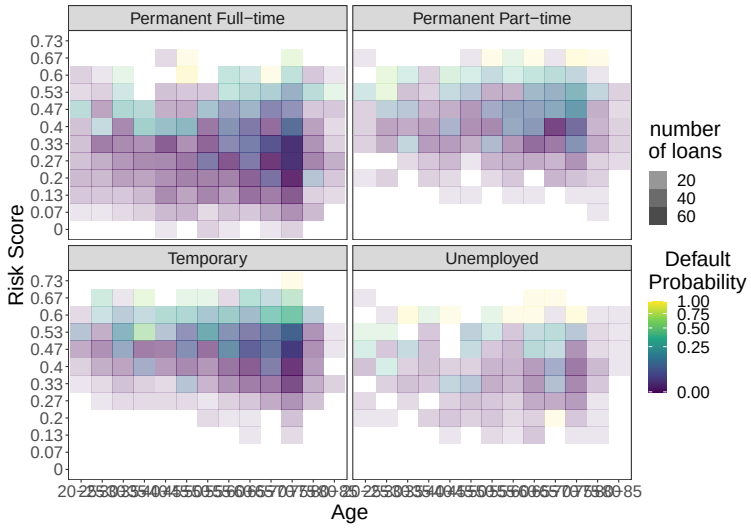
using code

```
ggplot(HW1Q5%>%mutate(lr=cut(loan.amount/yearly.income,breaks=(0:25)/25-1e-20),
                      cb=cut(credit.score,breaks=(0:30)*30-1e-20))%>%
       group_by(employment.status,lr,cb)%>%
       summarise(total=n(),def.prob=mean(default)),
       mapping=aes(y=lr,x=cb,alpha=total,fill=def.prob))+
    geom_tile()+
    largertextsize+
    scale_alpha_continuous(name="number\nof loans",trans="sqrt")+
    scale_fill_viridis_c(name="Default\nProbability",trans="sqrt")+
    scale_x_discrete(name="Credit Score",labels=(0:30)*30+15)+
    scale_y_discrete(name="Loan amount/Annual income",labels=(0:25)/25+0.02)+
    theme_bw()+
    theme(panel.grid.major=element_blank())+
    largertextsize+
    facet_wrap(employment.status~.)
```

To add age, we can replace the two predictors $\frac{\texttt{loan.amount}}{\texttt{yearly.income}}$ and `credit.score` by an overall risk score

$$\texttt{risk} = \frac{\texttt{loan.amount}}{\texttt{yearly.income}} - \frac{\texttt{credit.score}}{1000}$$

and plot a similar tile plot with this risk score and age on the axes:



using code

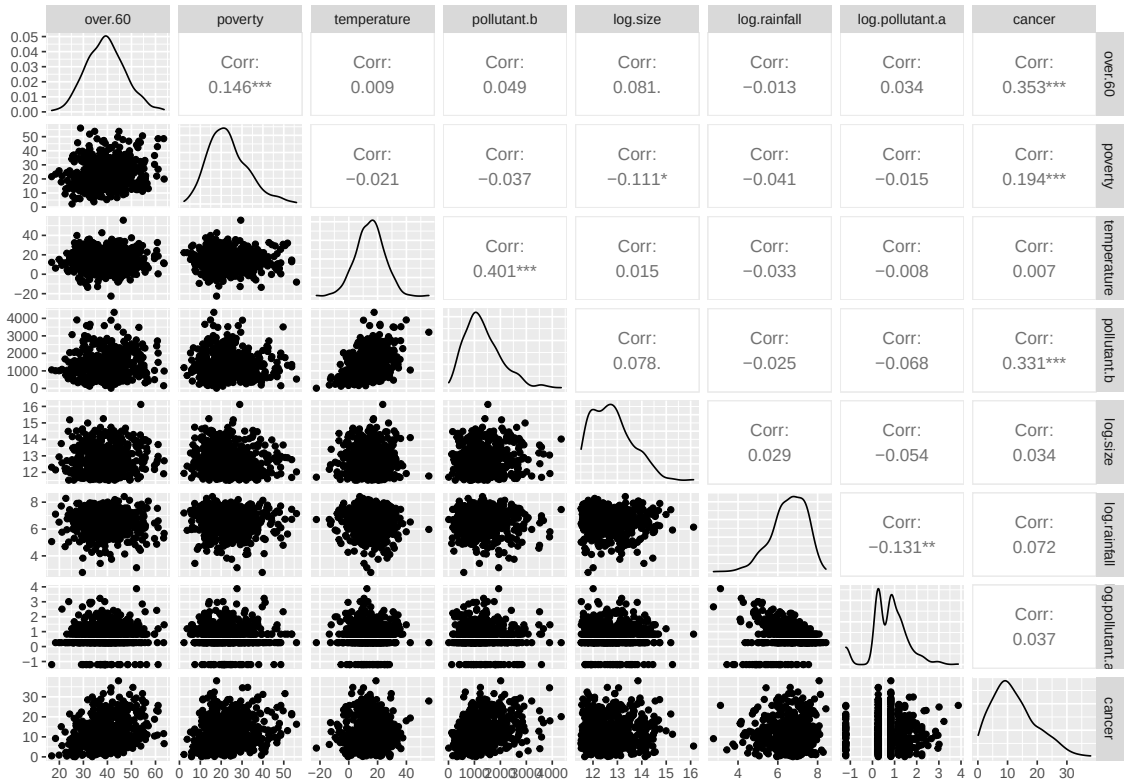
```
ggplot(HW1Q5%>%mutate(risk=loan.amount/yearly.income−credit.score/1000,
                      rc=cut(risk,breaks=(0:25)/15−1),
                      ac=cut(age,breaks=(0:15)*5+20))%>%
      group_by(employment.status,rc,ac)%>%
      summarise(total=n(),def.prob=mean(default)),
      mapping=aes(y=rc,x=ac,alpha=total,fill=def.prob))+
    geom_tile()+
    largertextsize+
    scale_alpha_continuous(name="number\nof loans",trans="sqrt")+
    scale_fill_viridis_c(name="Default\nProbability",trans="sqrt")+
    scale_x_discrete(name="Age",labels=paste((0:15)*5+20,"−",(0:15)*5+25,sep=""))+
    scale_y_discrete(name="Risk Score",labels=round((0:25)/15,2))+
    theme_bw()+
    theme(panel.grid.major=element_blank())+
    largertextsize+
    facet_wrap(employment.status~.)
```

6. *The file* `HW1Q6.txt` *contains data on the effect of pollution on cancer incidence.*

Variable name	*Meaning*
size	*The population of the town or city*
over.60	*The percentage of the population that are over 60.*
poverty	*The percentage of the population with income below the poverty level*
temperature	*The average temperature (°C) in the settlement.*
rainfall	*The average annual rainfall (mm).*
pollutant.a	*The average levels of pollutant a in the air (ppm)*
pollutant.b	*The average levels of pollutant b in the air (ppm)*
cancer.incidence	*The average annual number of new cancer diagnoses per 1000 residents.*

(a) Produce a figure to show these data for the purpose of data exploration.

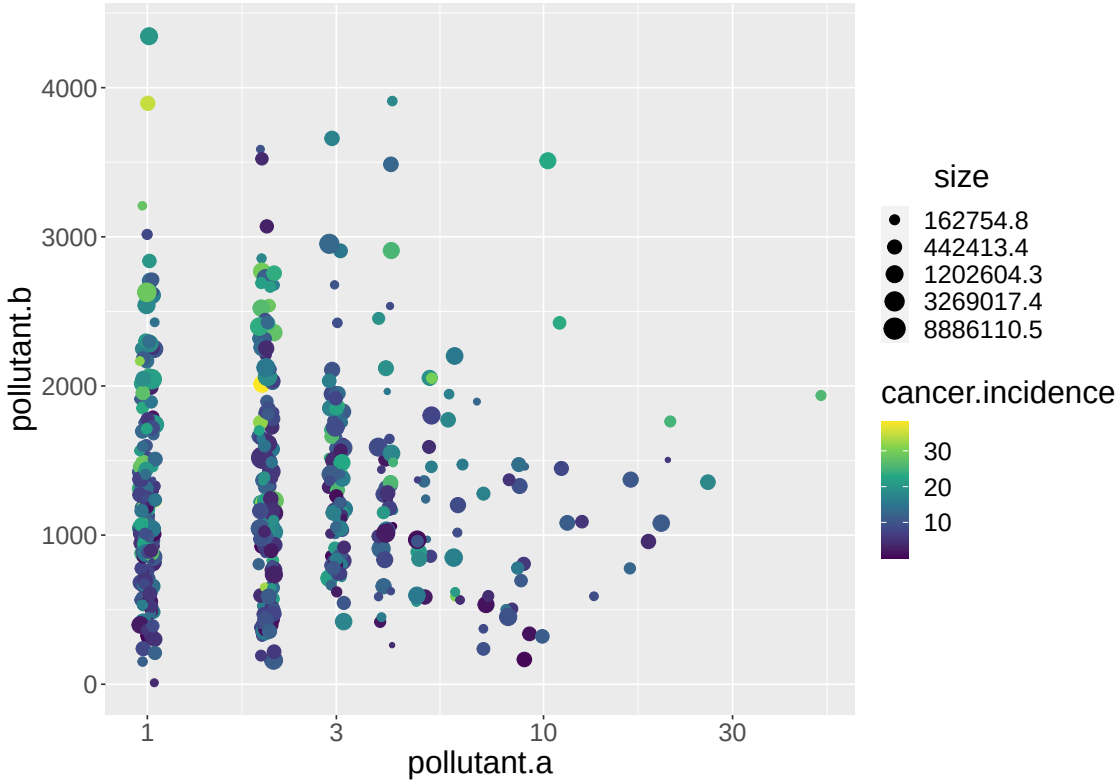
A first attempt is a collection of pairwise scatterplots for each pair of predictors, which can be produced using `ggpairs`.




```
ggpairs(HW1Q6%>%mutate(log.size=log(size),
                       log.rainfall=log(rainfall),
                       log.pollutant.a=log(pollutant.a+0.3),
                       cancer=cancer.incidence)%>%
        select(-c("size","rainfall","pollutant.a","cancer.incidence")))
```

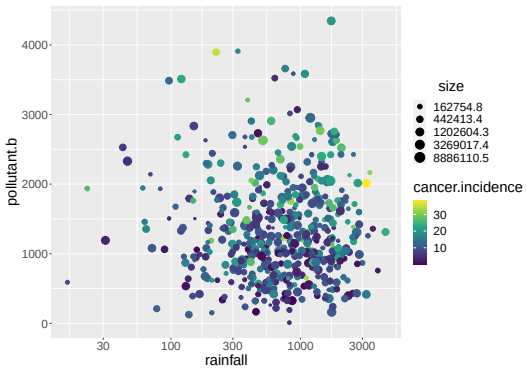
I have log-transformed the skewed variables `rainfall`, `pollutant.a` and `size`. I have added 0.3 to the `pollutant.a` prior to log-transformation to prevent taking a logarithm of 0. This choice is fairly arbitrary, and another choice could be made.

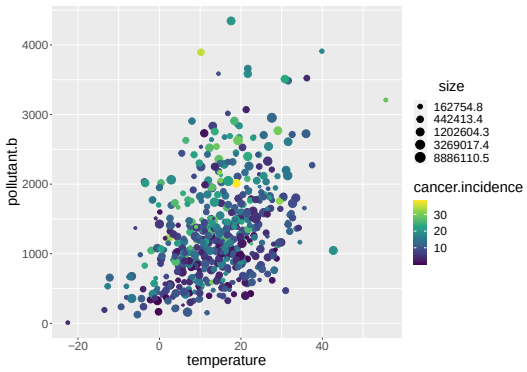
Since the focus is on the pollutants, another option is to use colour to represent `cancer.incidence` and use the x and y coordinates to represent the two pollutants, using size to represent population.




```
ggplot(HW1Q6, mapping=aes(colour=cancer.incidence,
                          x=pollutant.a,
                          y=pollutant.b,
                          size=size))+
    scale_x_log10()+
    geom_jitter(width=0.02,height=0)+
    scale_colour_viridis_c()+
    scale_size_continuous(trans="log")+
    largertextsize
```

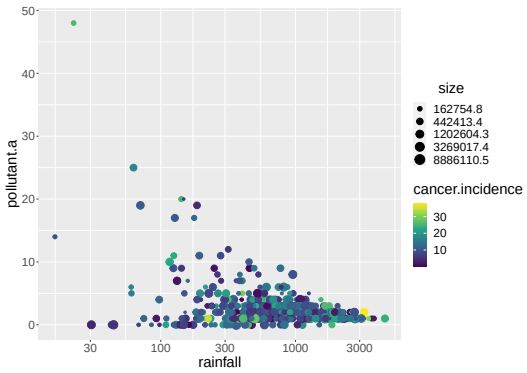
We can also show only one of the pollutants, and include rainfall or temperature:

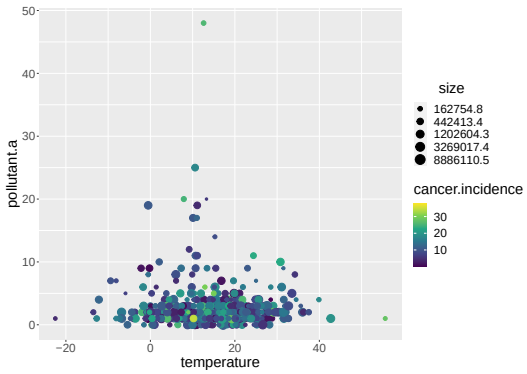





```
ggplot(HW1Q6, mapping=aes(colour=cancer.incidence,
                          x=rainfall,
                          y=pollutant.b,
                          size=size))+
    scale_x_log10()+
    geom_point()+
    scale_colour_viridis_c()+
    scale_size_continuous(trans="log")+
    largertextsize

ggplot(HW1Q6, mapping=aes(colour=cancer.incidence,
                          x=temperature,
                          y=pollutant.b,
                          size=size))+
    geom_point()+
    scale_colour_viridis_c()+
    scale_size_continuous(trans="log")+
    largertextsize
```

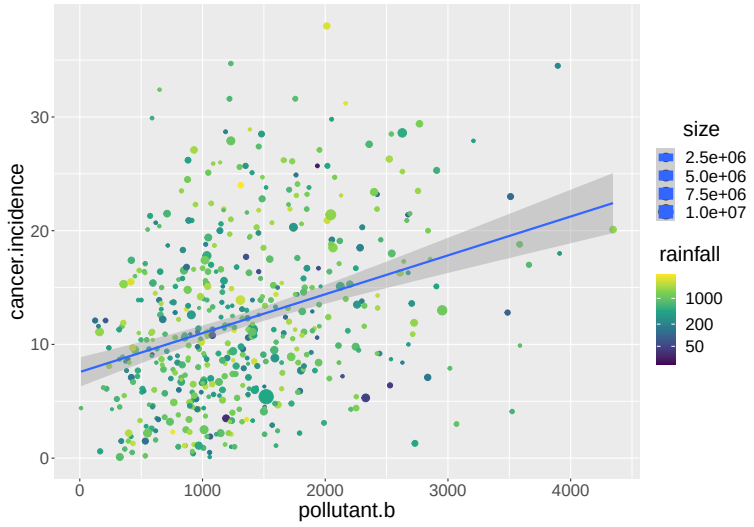
and similar plots for pollutant a.





(b) After analysing the data, you conclude that for fixed values of the other parameters, `pollutant.b` is positively associated with `cancer.incidence`, while for fixed values of the other parameters, `pollutant.a` is positively associated with `cancer.incidence` if `rainfall` is small. Make a plot which makes these conclusions more obvious.

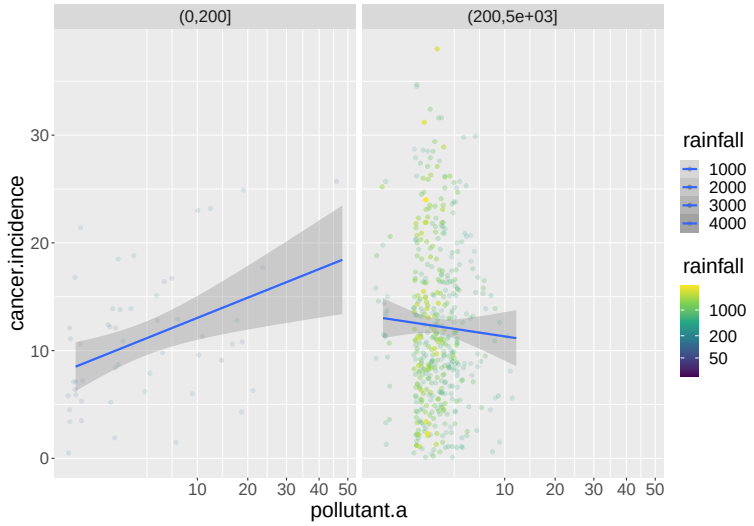
For the relation between `pollutant.b` and `cancer.incidence`, a simple scatterplot is sufficient.




```
ggplot(HW1Q6, mapping=aes(x=pollutant.b,
                          y=cancer.incidence,
                          colour=rainfall,
                          size=size))+
    geom_point()+
    scale_x_continuous()+
    scale_colour_viridis_c(trans="log",breaks=c(50,200,1000,5000))+
    geom_smooth(method="lm")+
    largertextsize
```

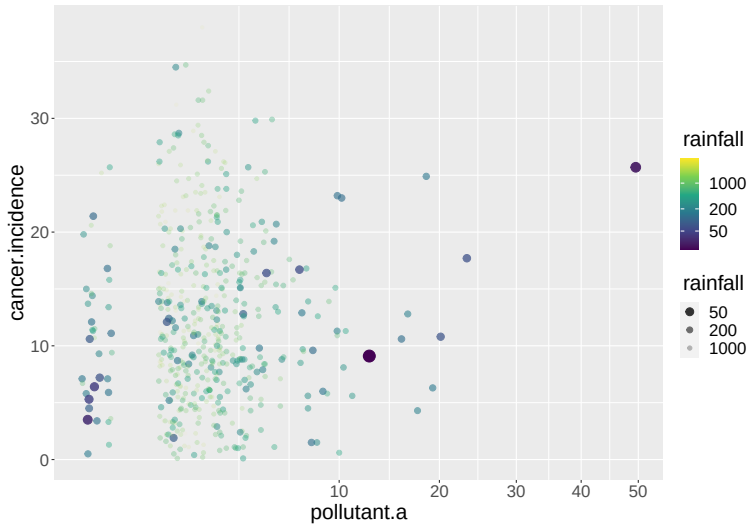
We could equally have chosen other predictors to be represented by colour and size.

For the relation between `pollutant.a` and `cancer.incidence`, the effect is only noticeable for small rainfall. Given the size of the effect and the negative correlation between pollutant a and rainfall, just colouring rainfall does not show the relation. Instead a `facet_wrap` with rainfall cut at 200 can show the pattern.




```
ggplot(HW1Q6, mapping=aes(x=pollutant.a, y=cancer.incidence, colour=rainfall))+geom_jitter(w
```

If we don't want to split the plot in a `facet_wrap`, we could try to highlight the points with low rainfall. For example, if we use reversed alpha and size scales for rainfall, then points with low rainfall will be larger and more opaque, so will be more visible, highlighting the pattern for low rainfall.



```
ggplot(HW1Q6, mapping=aes(x=pollutant.a,
                          y=cancer.incidence,
                          colour=rainfall,
                          alpha=rainfall,
                          size=rainfall))+
    geom_jitter(width=0.2,height=0)+
    scale_x_continuous(trans="sqrt")+
    scale_colour_viridis_c(trans="log",breaks=c(50,200,1000,5000))+largertextsize+
    scale_alpha_continuous(range=c(1,0),trans="log",breaks=c(0,50,200,1000,5000))+
    scale_size_continuous(range=c(5,1),trans="log",breaks=c(0,50,200,1000))
```