Lab 2: Descriptive Satistics and ggplot2

Hao Wang

2017 Fall

```
## [1] "/Users/haowang/Dropbox/2017 Spring/ 603/lab/Lab 2"
```

Working Example: Baylor's American Religious Survey

The following R chunk load data from Baylor Regilious Survey. Baylor Religious Study is a comprehensive analysis on religious beliefs in the United States. For detailed explanation please refer to http://www.thearda.com/Archive/Files/Descriptions/BAYLORW2.asp. Total sample size is 1648 adults, with 318 variables, survey was conducted by Gallup.

```
#The first step is to load Byalor Religious Survey 2005 (I)
mydata <- read.dta13("http://www.thearda.com/download/download.aspx?file=Baylor%20Religion%20Survey,%20U
```

Subsetting Data

Because the raw file includes some missing points, I create a new dataset of potential interesting variables without missing data.

```
plotdata <- mydata[, c('relgious', 'attend', 'gaymarr', 'gayborn',
                       'votefem', 'partyid')]
#religious: how religious you are
#attend:    how often do you attend church
#gaymarr:   gay marriage should be legal
#gayborn:   people are born either homo or heterosexual
#vetefem:   would you vote for a female cadidate nominated by your party
#partyid:   democract or republican
plotdata <- na.omit(plotdata)
```

Mean attitudes towards gay marriage

```
summary(plotdata$gaymarr)
```

```
## Strongly disagree          Disagree             Agree    Strongly agree
##               507               311               223               291
##         Undecided
##               159
```

```
mean(plotdata$gaymarr)
```

```
## Warning in mean.default(plotdata$gaymarr): argument is not numeric or
## logical: returning NA
```

```
## [1] NA
```

What happened? we got an error message since all the survey items are coded in the categorical way.

Convert categorical data into numeric

```
plotdata2=as.data.frame(sapply(plotdata, as.numeric))#sapply function returns matrix with same
#legnth, and in the same time converted factors into numeric numbers by as.numeric command
table(plotdata2$gaymarr)
```

```
##
##   1   2   3   4   5
## 507 311 223 291 159
```

```
mean(plotdata2$gaymarr)
```

```
## [1] 2.519785
```

- Lab Practice 1: How to interpret the mean value of plotdata2? Is it correct?

Recode variables

```
plotdata2$gaymarr1 <- plotdata2$gaymarr
plotdata2$gaymarr1[plotdata2$gaymarr1 == 5] <- NA
mean(plotdata2$gaymarr1, na.rm = TRUE)
```

```
## [1] 2.223724
```

```
median(plotdata2$gaymarr1, na.rm = TRUE)
```

```
## [1] 2
```

Get an overview by the summary() command

```
summary(plotdata)
```

```
##                  relgious                      attend
##  Not at all religious:165   Never                :327
##  Not too religious   :213   Weekly               :318
##  Somewhat religious  :617   Several times a year :168
##  Very religious      :483   Once or twice a year :155
##  Don't know          : 13   Several times a week :137
##                             Less than once a year:122
##                             (Other)              :264
##               gaymarr                  gayborn    votefem
##  Strongly disagree:507   Strongly disagree:254   Yes:1190
##  Disagree         :311   Disagree         :259   No : 301
##  Agree            :223   Agree            :420
##  Strongly agree   :291   Strongly agree   :281
```

```
##  Undecided        :159   Undecided        :277
##
##
##                 partyid
##  Independent        :311
##  Moderate Democrat  :281
##  Moderate Republican:261
##  Strong Democrat    :196
##  Strong Republican  :174
##  Leaning Republican :125
##  (Other)            :143
```

Frenquency Table

```
table(plotdata$gaymarr)
```

```
##
## Strongly disagree          Disagree             Agree    Strongly agree
##               507               311               223               291
##         Undecided
##               159
```

Graphing Using ggplot2

The ggplot2 package, created by Hadley Wickham, offers a powerful graphics language for creating elegant and complex plots. Its popularity in the R community has exploded in recent years. Origianlly based on Leland Wilkinson's The Grammar of Graphics, ggplot2 allows you to create graphs that represent both univariate and multivariate numerical and categorical data in a straightforward manner. Grouping can be represented by color, symbol, size, and transparency. The creation of trellis plots (i.e., conditioning) is relatively simple.

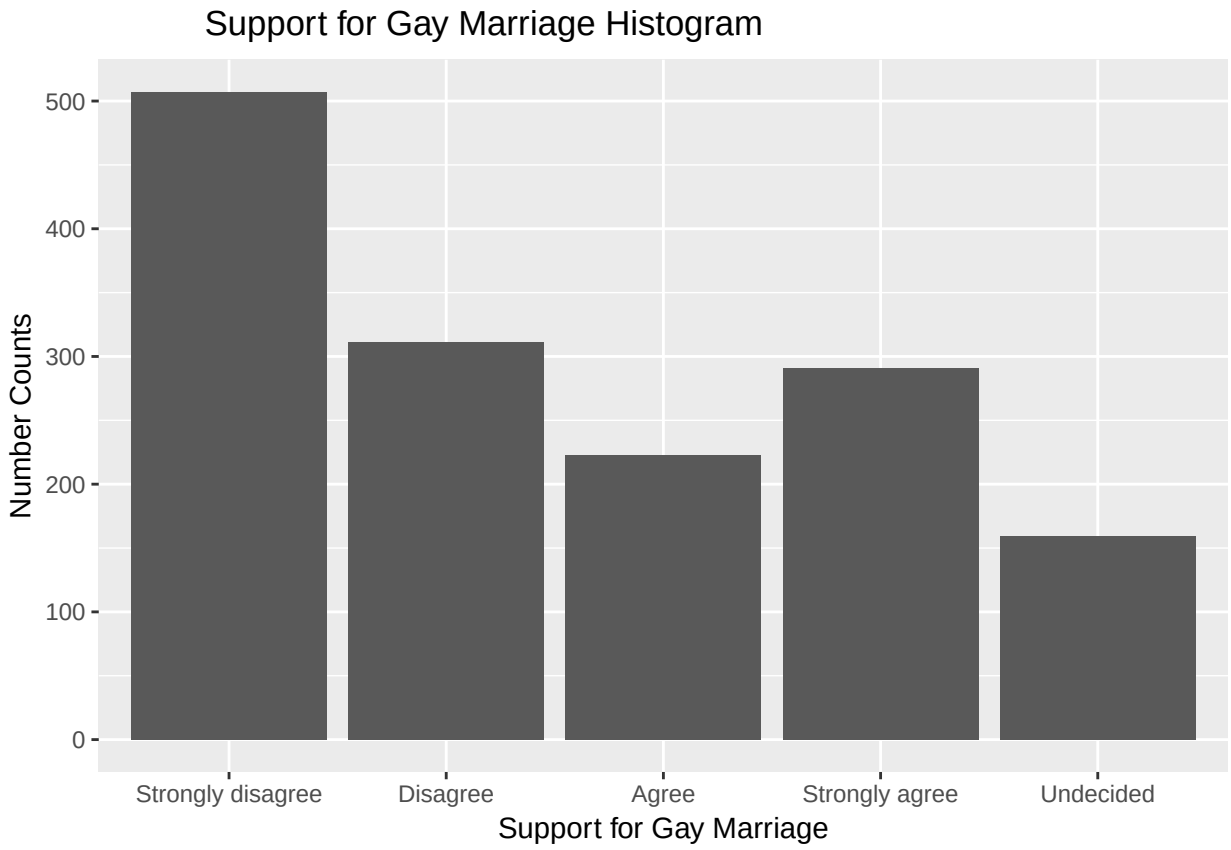
ggplot 2 reference guide: http://docs.ggplot2.org/current/index.html#

ggplot 2 cheatsheet: https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf

Histogram

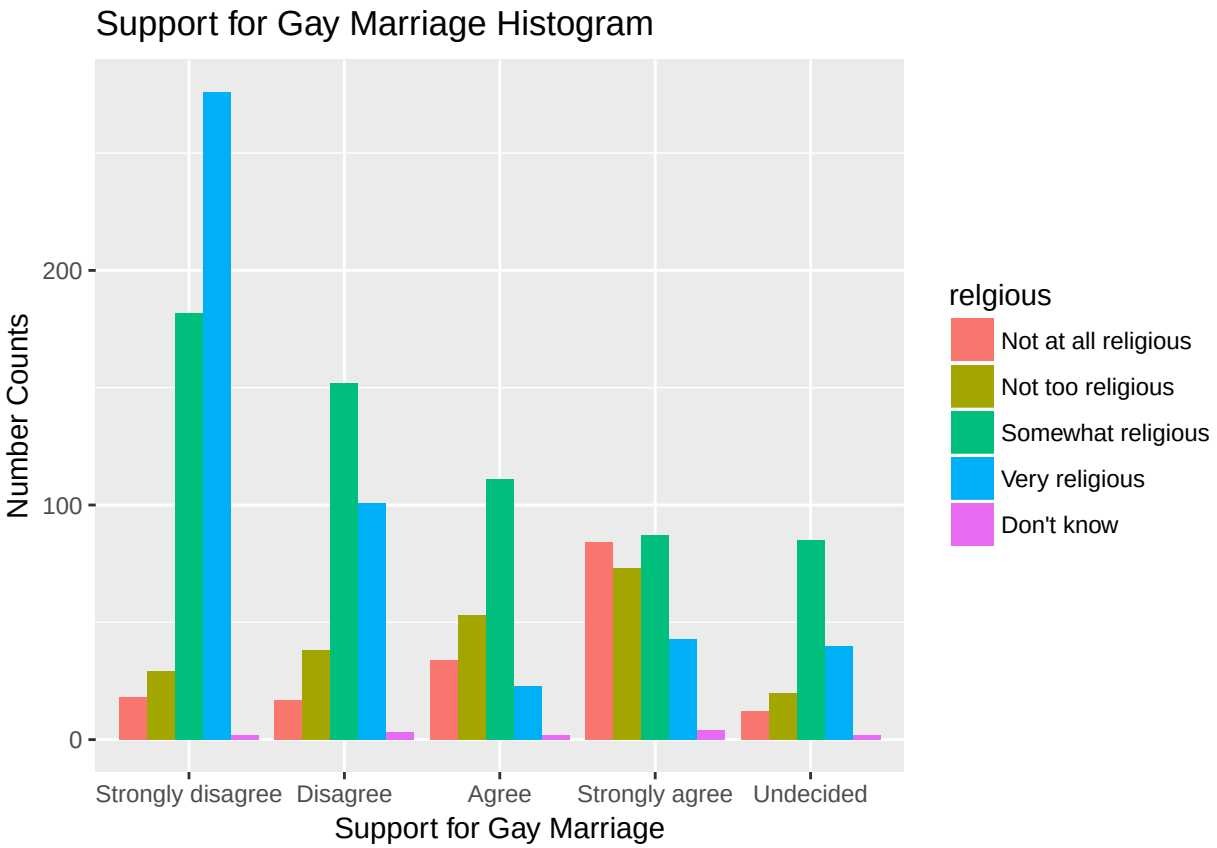
This show the histogram of Attitudes on gay marriage

```
library(ggplot2)
hist.gay <- ggplot(plotdata, aes(x = gaymarr)) +
            geom_histogram(stat="count", alpha =1) +
            labs(x="Support for Gay Marriage", y="Number Counts") +
            ggtitle("           Support for Gay Marriage Histogram")
hist.gay
```



We can also plot the histogram according to religious degrees.

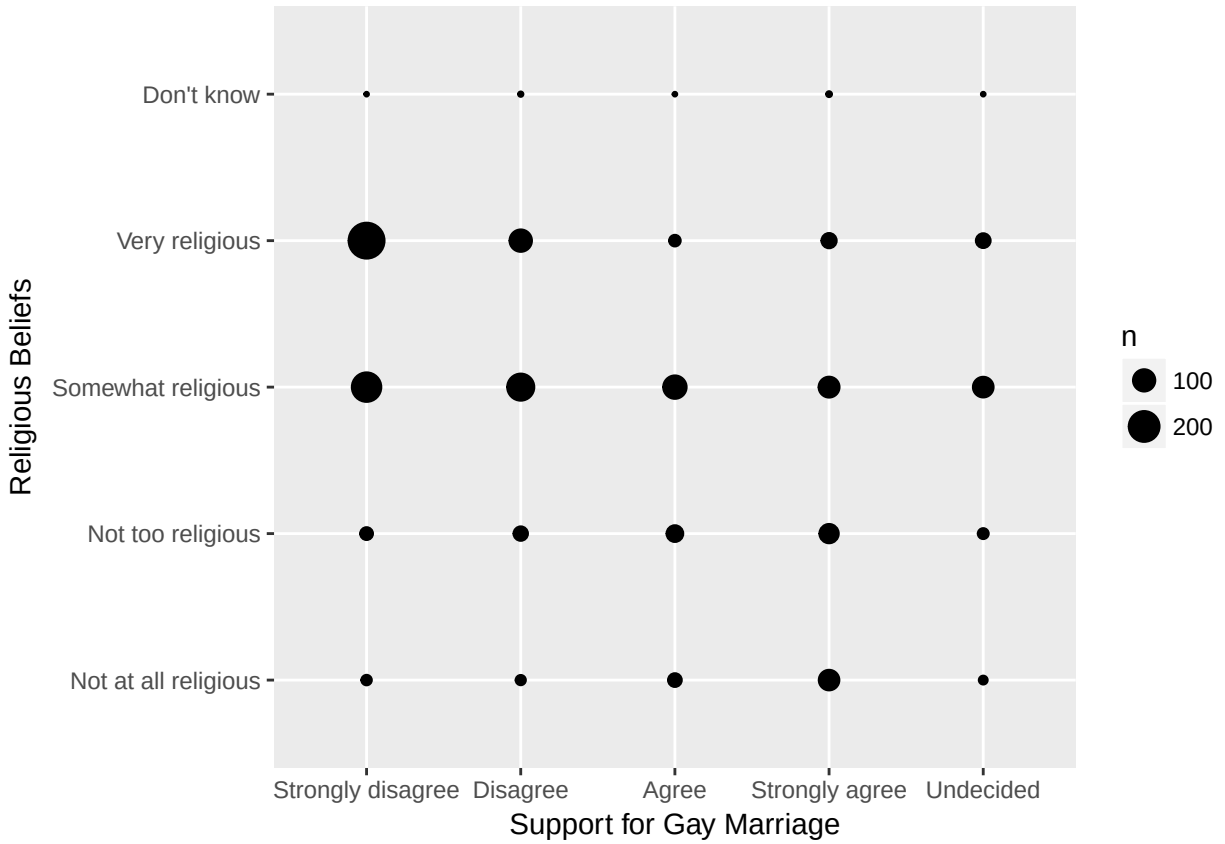
```
hist.relig.gay <- ggplot(plotdata, aes(x =gaymarr, fill=relgious)) +
                  geom_histogram(stat="count",alpha =1, position = "dodge")+
                  ylab("Number Counts") +
                  xlab("Support for Gay Marriage")+
                  ggtitle("Support for Gay Marriage Histogram")
hist.relig.gay
```



Scatter Plot

In scatter plot we write two parameters in the aes() option. I use additional option geom_count() here to illustrate the size.

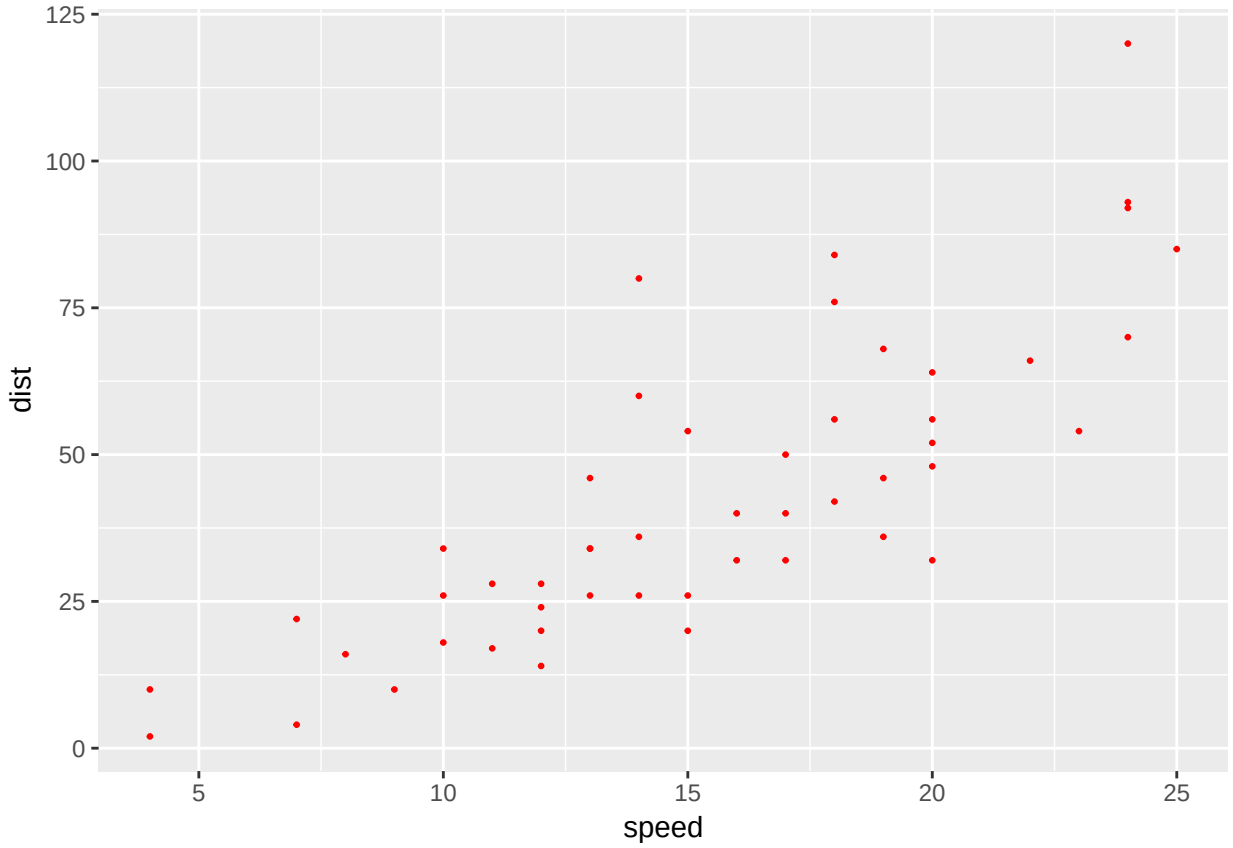
```
relig.gaymarr <- ggplot(data=plotdata, aes(y = relgious, x = gaymarr)) +
                 geom_count() +
                 scale_size_area() +
                 ylab("Religious Beliefs") +
                 xlab("Support for Gay Marriage")
relig.gaymarr
```



Adding lines to scatter plot

Let's use the default data from the car package here. It measures the relation between car speed and stop distance.

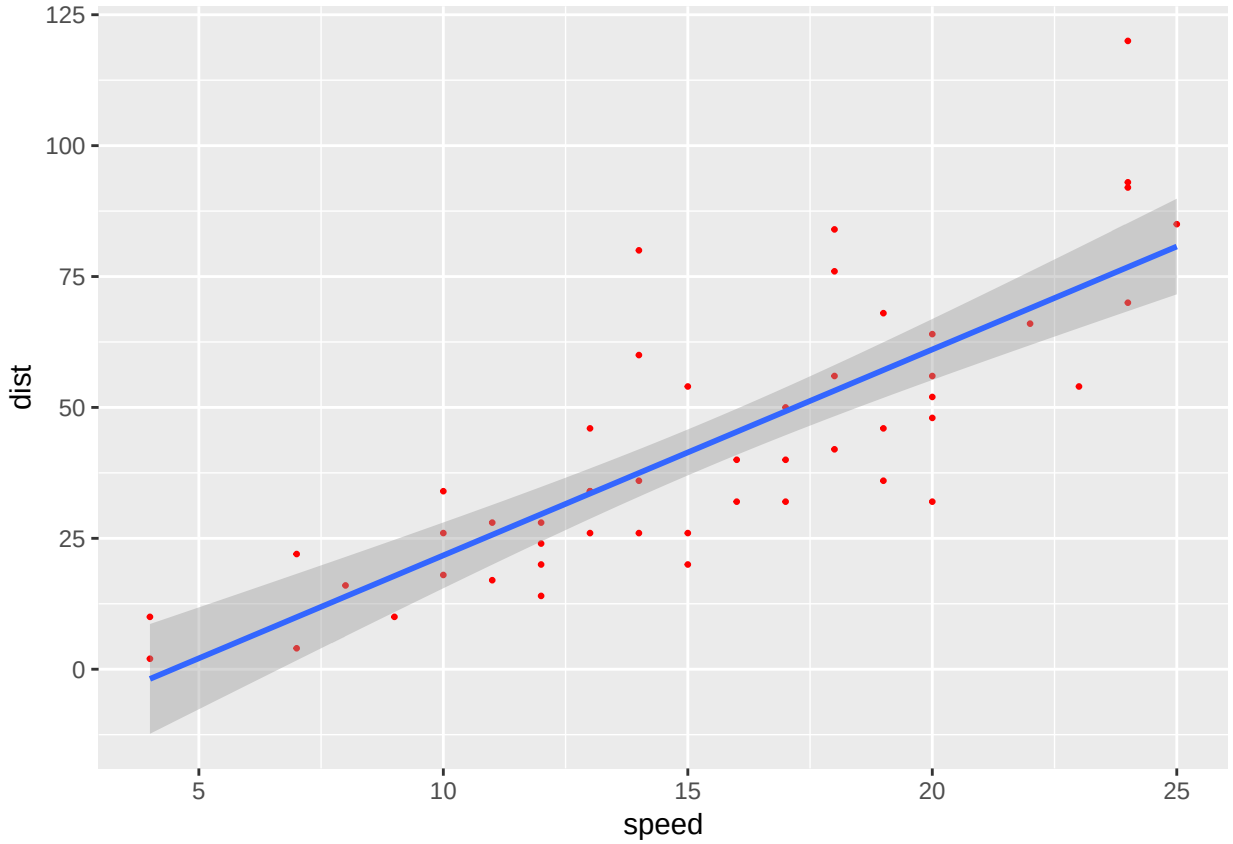
```
mydata <- cars
car <- ggplot(data = mydata, aes(x = speed, y = dist)) +
       geom_point(size=0.5, color = "red")
car
```



Let's adding a line.

Method 1, use the default geom_smooth()

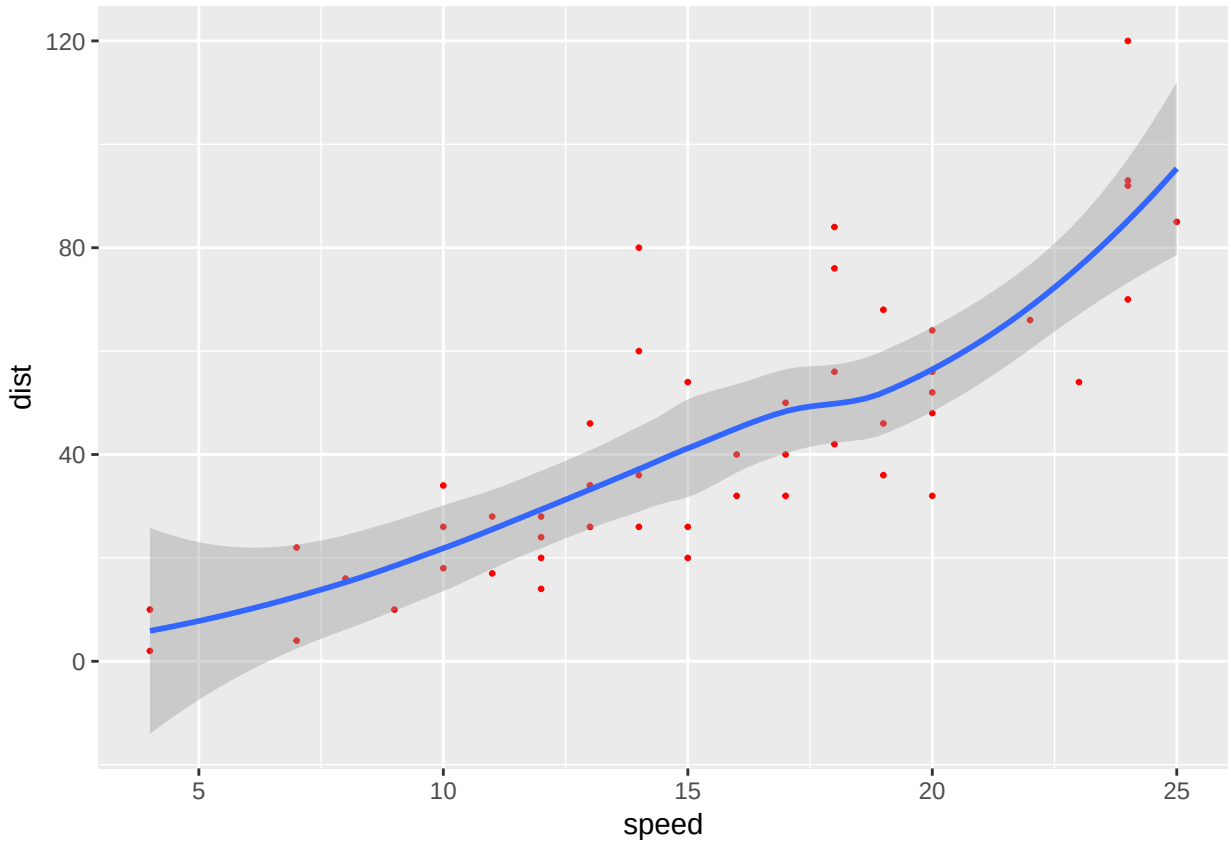
```
car <- ggplot(data = mydata, aes(x = speed, y = dist)) +
       geom_point(size=0.5, color = "red") +
       geom_smooth(method = 'lm', se=TRUE)
car
```



You can try other options of geom_smooth()

LOESS is a nonparametric method that combine multiple regression models in a k-nearest-neighbor-based modeling.

```
car <- ggplot(data = mydata, aes(x = speed, y = dist)) +
       geom_point(size=0.5, color = "red") +
       geom_smooth(method = 'loess', se=TRUE)
car
```



If you want full control over your line

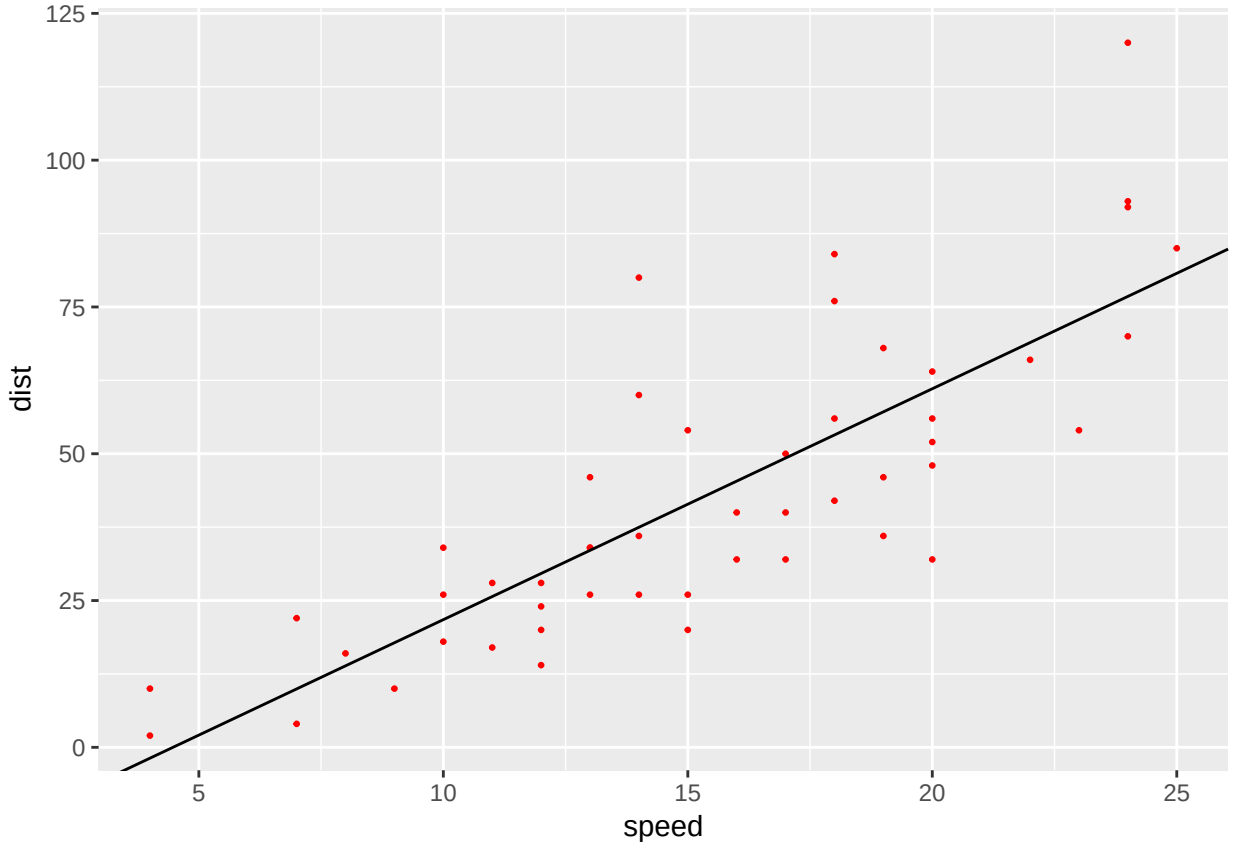
In this condition you need to calculate all the parameters of the line. Let's try with linear models

```
lm <- lm(dist ~ speed, data = mydata)
summary(lm)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = mydata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

```
intercept <- summary(lm)$coefficients[1,1]
slope <- summary(lm)$coefficients[2,1]

car <- ggplot(data = mydata, aes(x = speed, y = dist)) +
       geom_point(size=0.5, color = "red") +
       geom_abline(intercept = intercept, slope = slope)
car
```



The summary(lm) command returns coefficients of the regression. In this case we need to extract intercept in row 1, column 1; and slope in row 2, column 1.