# POLS 503, Spring 2016: Assignment 1

## Xingwei Wu

### Problem 1: Data Wrangling and Viz Refresher

a. Load the democracy data frame

```
democracy <- read.csv(file = "democracy.csv", stringsAsFactors = FALSE,na.strings = ".")
```

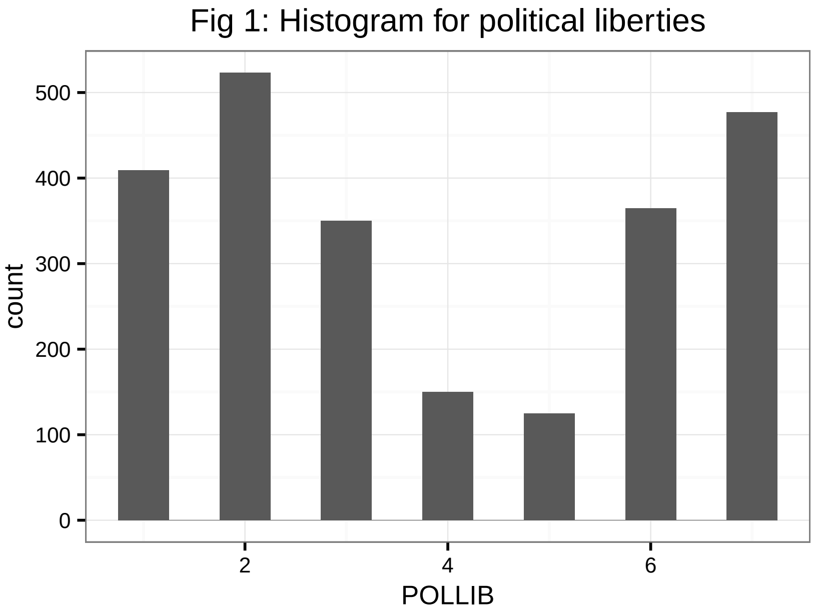
b. Create data frame with statistics (means, medians, and ) for all variables

```
democracy_by_variable <-
   democracy %>%
   gather(variable,value,-COUNTRY, -CTYNAME, -REGION, -YEAR)
summary<-democracy_by_variable%>%
  na.omit() %>%
  group_by(variable) %>%
  summarise(min=min(value),
            mean = mean(value),
            sd = sd(value),
            max = max(value))
knitr::kable(summary,align="c")
```

| variable | min | mean | sd | max |
|:-:|:-:|:-:|:-:|:-:|
| BRITCOL | 0.00 | 0.2433349 | 0.4291476 | 1.00 |
| CATH | 0.00 | 37.1991275 | 38.2807940 | 99.00 |
| CIVLIB | 1.00 | 4.0762818 | 1.9732405 | 7.00 |
| EDT | 0.03 | 4.8533925 | 3.1173053 | 12.81 |
| ELF60 | 0.00 | 0.3994587 | 0.2965374 | 0.93 |
| GDPW | 480.00 | 8876.9592826 | 8016.9287248 | 37903.00 |
| MOSLEM | 0.00 | 19.7358943 | 34.0430019 | 99.90 |
| NEWC | 0.00 | 0.4561318 | 0.4981322 | 1.00 |
| OIL | 0.00 | 0.1000969 | 0.3001656 | 1.00 |
| POLLIB | 1.00 | 3.8595248 | 2.2326768 | 7.00 |
| REG | 0.00 | 0.3986912 | 0.4896883 | 1.00 |
| STRA | 0.00 | 0.3751818 | 0.6979135 | 5.00 |

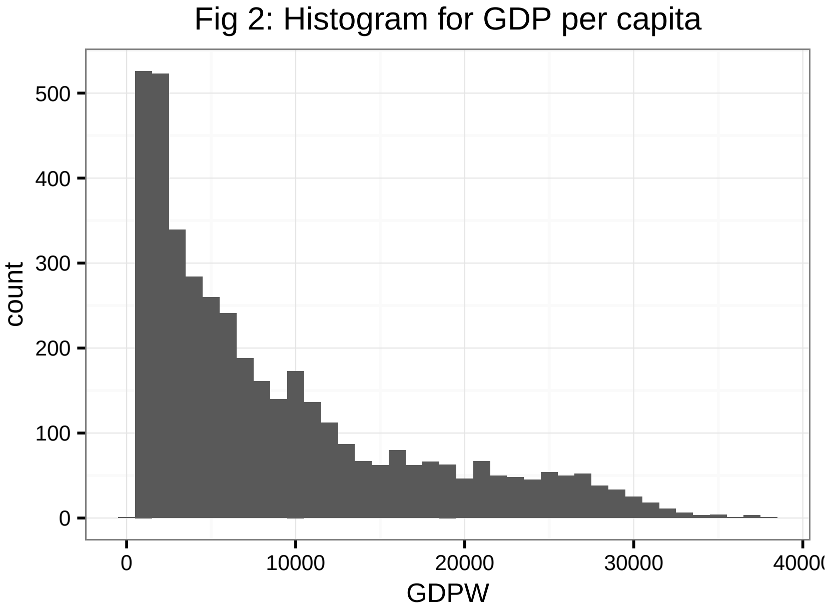
d. Histogram for political liberties

```
ggplot(democracy,
       aes(x = POLLIB)) +
       geom_histogram(binwidth=.5) +
       ggtitle("Fig 1: Histogram for political liberties")+
       theme_bw()+
       theme(text=element_text(size=10),
       axis.title=element_text(size=10))
```



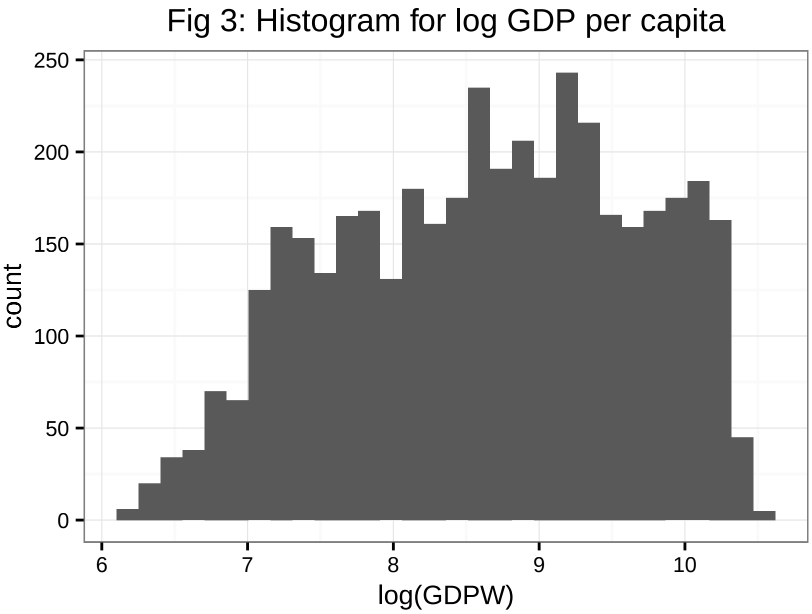
e. Histogram for GDP per capita

```
ggplot(democracy,
       aes(x = GDPW)) +
       geom_histogram(binwidth = 1000) +
       ggtitle("Fig 2: Histogram for GDP per capita")+
       theme_bw()+
       theme(text=element_text(size=10),
       axis.title=element_text(size=10))
```



f. Histogram for log GDP per capita

```
ggplot(democracy,
       aes(x = log(GDPW))+
       geom_histogram() +
       ggtitle("Fig 3: Histogram for log GDP per capita")+
       theme_bw()+
       theme(text=element_text(size=10),
       axis.title=element_text(size=10))
```

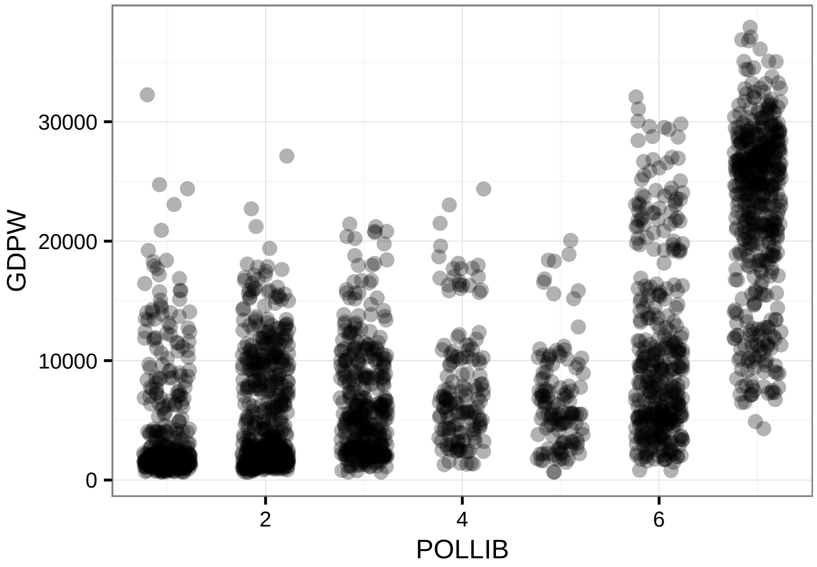


From the histograms of GDP and logGDP per capita, the distrbution of GDP per capita is similar to exponential distribution but the log GDP per capita is more like normal distribution.

g. Plot political liberties against GDP per capita

```
ggplot(democracy,aes(POLLIB, GDPW))+
  geom_point(position = position_jitter(width = 0.6),alpha=0.3,size=2)+
  ggtitle("Fig 4: Political liberties vs.GDP per capita")+
  theme_bw()+
  theme(text=element_text(size=10),
        axis.title=element_text(size=10),
        legend.position="bottom",
        legend.text=element_text(size=10))
```

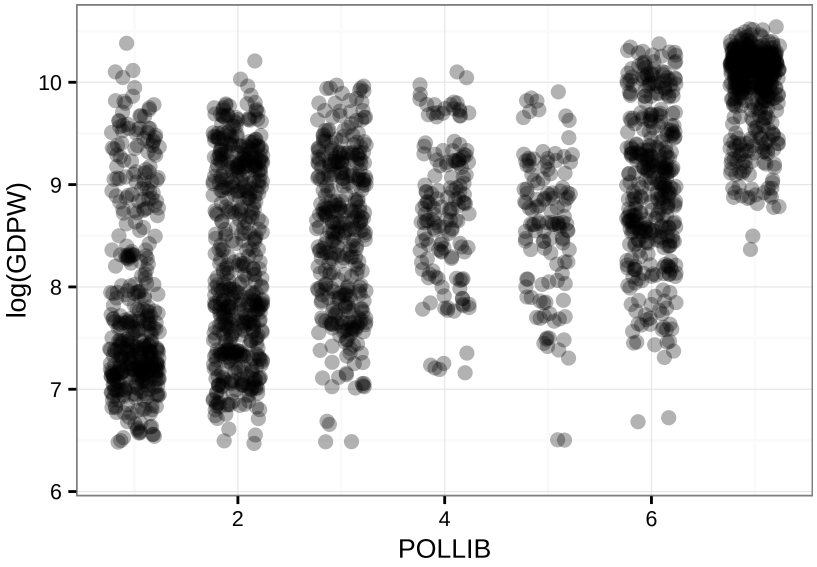
Fig 4: Political liberties vs.GDP per capita



i. Plot political liberties against log GDP per capita

```r
ggplot(democracy,aes(POLLIB, log(GDPW)))+
  geom_point(position = position_jitter(width = 0.6),alpha=0.3,size=2,na.rm=TRUE)+
  ggtitle("Fig 5: Political liberties vs.GDP per capita")+
  theme_bw()+
  theme(text=element_text(size=10),
        axis.title=element_text(size=10),
        legend.position="bottom",
        legend.text=element_text(size=10))
```

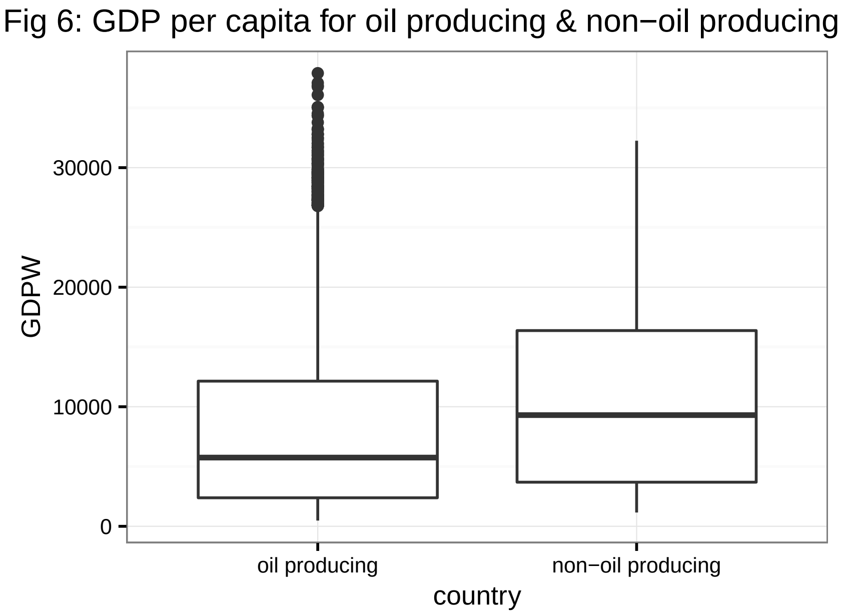
Fig 5: Political liberties vs.GDP per capita



From the distributions of political liberties against GDP and log GDP per capita, GDP for countries with highest political liberty (7) is much higher than those with lower political liberty (1-6). There is very slite increasing trend of GDPs in countries with political liberty degree of 1-6. But with logged GDP, the gradient increasing trend becomes clearer for countries with political liberty degree from 1 to 7.

j. Boxplot of GDP per capita for oil producing and non-oil producing nations

```
democracy$OIL<-factor(democracy$OIL,levels=c("0","1"),labels=c("oil producing","non-oil producing"))
ggplot(democracy,aes(OIL, GDPW))+
  geom_boxplot()+
  ggtitle("Fig 6: GDP per capita for oil producing & non-oil producing nations")+
  theme_bw()+
  xlab("country")+
  theme(text=element_text(size=10),
        axis.title=element_text(size=10),
        legend.position="bottom",
        legend.text=element_text(size=10))
```



k. Mean GDP per capita in countries with at least 40 percent Catholics and all countries
   The mean GDP per capita in countries with at least 40 percent Catholics is 10295, which higher than that in all countries (8877)

```
print(list(summarize(democracy,
            GDP_mean_all=mean(GDPW)),
            summarize(filter(democracy,CATH>=40),
            GDP_mean_40=mean(GDPW))))
```

```
## [[1]]
##   GDP_mean_all
## 1     8876.959
##
## [[2]]
##   GDP_mean_40
## 1    10295.13
```

l. Average GDP per capita in countries with different ethnolinguistic fractionalization group

```
democracynew<- mutate(democracy,Country_ELF60 = ifelse(is.na(ELF60)==TRUE,"Missing", ifelse(ELF60<0.6,"
knitr::kable(democracynew%>%
  group_by(Country_ELF60)%>%
  summarize(GDP_mean=mean(GDPW)),align="c")
```

| Country_ELF60 | GDP_mean |
|:---:|:---:|
| Greater than 60% | 3590.939 |
| Less than 60% | 11803.780 |
| Missing | 7767.245 |

m. Median average years of education for all years

```
EDT_median1<-democracy%>%
  group_by(YEAR)%>%
  filter(is.na(EDT)==F) %>%
  summarize(EDT_median=median(EDT))
knitr::kable(head(EDT_median1),align="c")
```

| YEAR | EDT_median |
|:---:|:---:|
| 1960 | 4.4600 |
| 1961 | 2.6300 |
| 1962 | 2.6850 |
| 1963 | 2.7850 |
| 1964 | 2.9025 |
| 1965 | 3.0150 |

```
ggplot(democracy, aes(YEAR, EDT)) +
  geom_point(alpha=0.3,na.rm = FALSE)+
  stat_summary(fun.y = median, geom="line",colour="red",lwd=1)+
  xlim(c(1955,1990))+
  ggtitle("Fig 7: Median average years of education ")+
  theme_bw()+
  theme(text=element_text(size=10),
        axis.title=element_text(size=10))
```

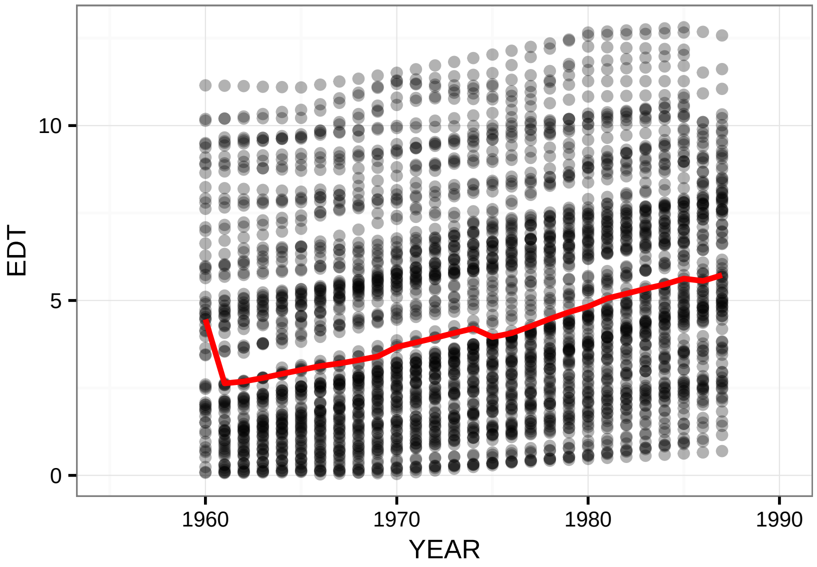


Fig 7: Median average years of education

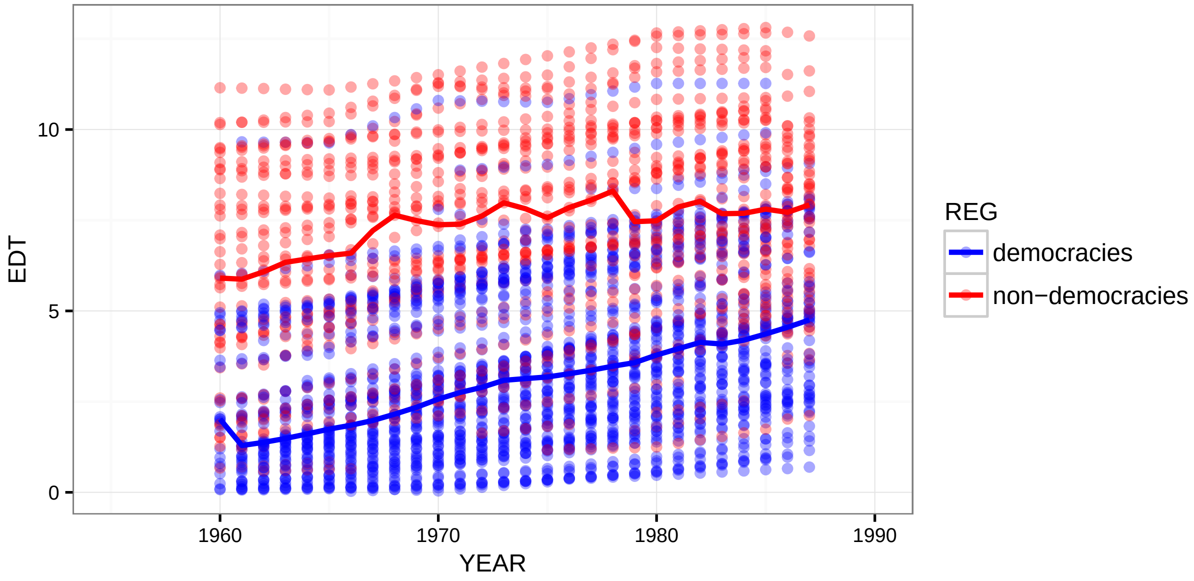
o. Median average years of education group by both year and democracy

```
EDT_median2<-democracy%>%
  group_by(YEAR,REG)%>%
  filter(is.na(EDT)==F) %>%
  summarize(EDT_median=median(EDT))
knitr::kable(head(EDT_median2),align="c")
```

| YEAR | REG | EDT_median |
|:---:|:---:|:---:|
| 1960 | 0 | 2.0200 |
| 1960 | 1 | 5.9050 |
| 1961 | 0 | 1.2900 |
| 1961 | 1 | 5.8750 |
| 1962 | 0 | 1.3775 |
| 1962 | 1 | 6.0800 |

```
ggplot(democracy, aes(YEAR,EDT)) +
  aes(colour = factor(REG))+
  geom_point(alpha=0.35) +
  stat_summary(fun.y = median, geom="line",lwd=1)+
  scale_color_manual("REG ",labels = c("democracies", "non-democracies"),values = c("blue", "red"))+
  xlim(c(1955,1990))+
  theme_bw()+
  ggtitle("Fig 8: Average education years for democracies vs.non-democracies")+
  theme(text=element_text(size=10),
        axis.title=element_text(size=10),
        legend.position="right",
        legend.text=element_text(size=10))
```

Fig 8: Average education years for democracies vs.non−democracies



n. Country closest to the median years of education in 1985
Venezuela is the country which had the closest years of education to the median in 1985.

```
democracy_1985<-democracy%>%
  filter(YEAR==1985&is.na(EDT)==F)%>%
  mutate(Difference = abs(EDT-median(EDT)))

Unique<-democracy_1985%>%
  arrange(Difference)%>%
  slice(1)
print(c(Unique$CTYNAME,Unique$EDT))
```

```
## [1] "Venezuela" "5.625"
```

q. 25th and 75th percentiles of ethnolinguistic fractionalization for new and old countries

```
ELF60_p2575<-democracynew%>%
  filter(is.na(ELF60)==F)%>%
  group_by(NEWC)%>%
  summarize(ELF60_p25=quantile(ELF60,0.25),
            ELF60_p75=quantile(ELF60,0.75))
knitr::kable(ELF60_p2575,align="c")
```

| NEWC | ELF60_p25 | ELF60_p75 |
|:---:|:---:|:---:|
| 0 | 0.06 | 0.44 |
| 1 | 0.42 | 0.75 |

### Problem 2: Plotting data and regressions

a. Statistic summary of x and y in each dataset The mean and standard deviations of x and y, and correlation between x and y as well as the linear regression between x and y for each dataset are shown in the following tables.

```
anscombe2 <- anscombe %>%
    mutate(obs = row_number()) %>%
    gather(variable_dataset, value, - obs) %>%
    separate(variable_dataset, c("variable", "dataset"), sep = 1L) %>%
    spread(variable, value) %>%
    arrange(dataset, obs)

options(digits = 3)
summaryxy<-anscombe2 %>%
  group_by(dataset) %>%
  summarise(x_mean = mean(x),
            x_sd = sd(x),
            y_mean = mean(y),
            y_sd = sd (y),
            xy_correlation=cor(x,y))
knitr::kable(summaryxy,align = 'c')
```

| dataset | x_mean | x_sd | y_mean | y_sd | xy_correlation |
|:---:|:---:|:---:|:---:|:---:|:---:|
| 1 | 9 | 3.32 | 7.5 | 2.03 | 0.816 |
| 2 | 9 | 3.32 | 7.5 | 2.03 | 0.816 |
| 3 | 9 | 3.32 | 7.5 | 2.03 | 0.816 |
| 4 | 9 | 3.32 | 7.5 | 2.03 | 0.817 |

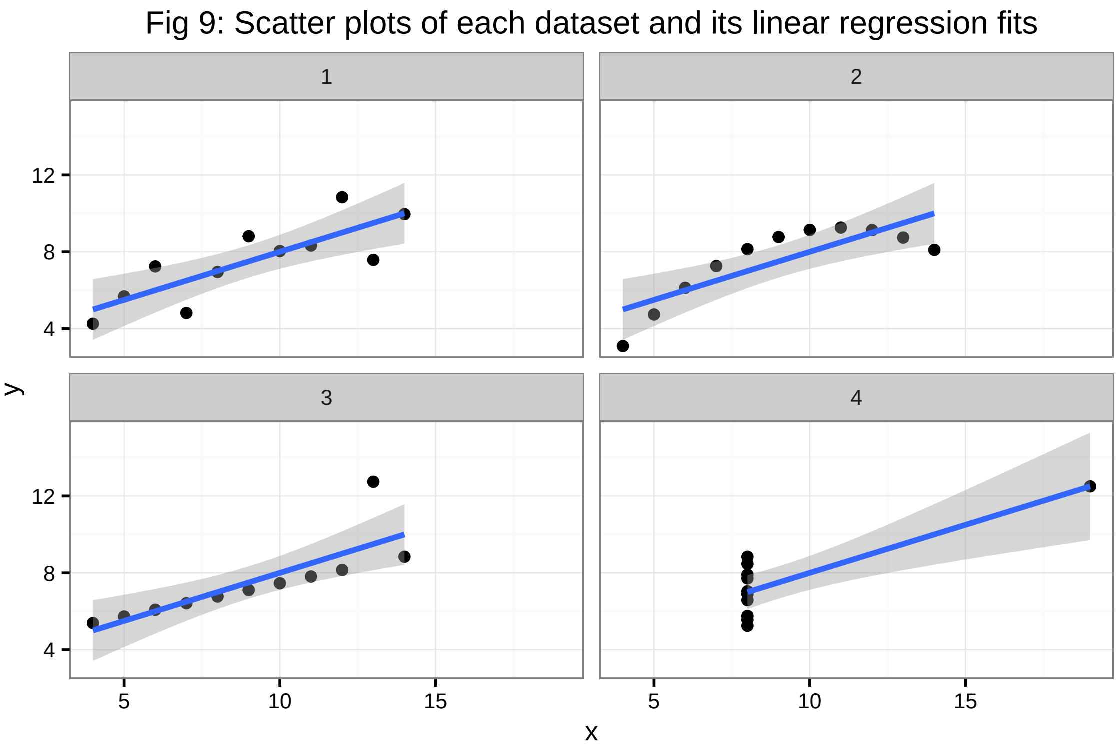
```
regressionxy<-anscombe2%>%
   group_by(dataset) %>%
   do(tidy(lm(.$y ~ .$x)))
knitr::kable(regressionxy,align = 'c')
```

| dataset | term | estimate | std.error | statistic | p.value |
|:---:|:---:|:---:|:---:|:---:|:---:|
| 1 | (Intercept) | 3.0 | 1.125 | 2.67 | 0.026 |
| 1 | .$x | 0.5 | 0.118 | 4.24 | 0.002 |
| 2 | (Intercept) | 3.0 | 1.125 | 2.67 | 0.026 |
| 2 | .$x | 0.5 | 0.118 | 4.24 | 0.002 |
| 3 | (Intercept) | 3.0 | 1.124 | 2.67 | 0.026 |
| 3 | .$x | 0.5 | 0.118 | 4.24 | 0.002 |
| 4 | (Intercept) | 3.0 | 1.124 | 2.67 | 0.026 |
| 4 | .$x | 0.5 | 0.118 | 4.24 | 0.002 |

These four dataset have almost the same means and standard deviations of x and y as well as correlation coefficients between x and y. The regression between y and x for each dataset are also very similar. Hence, we infer these datasets look very similar.

b. Scatter plots of each dataset and its linear regression fits

```
ggplot(anscombe2, aes(x, y)) +
  geom_point() +
  geom_smooth(method="lm") +
  facet_wrap(~ dataset,2,2)+
  ggtitle("Fig 9: Scatter plots of each dataset and its linear regression fits")+
  theme_bw()+
  theme(text=element_text(size=10),
        axis.title=element_text(size=10),
        legend.position="bottom",
        legend.text=element_text(size=10))
```



The scatter plots for each dataset showed that the datasets are different.

### Problem 3: Predicting Sprint Times

a. Create the new dataset named `sprinters_orig` with observations only from plympics.

```
sprinters <- read.csv("sprinters.csv")
```

```
sprinters_orig <-filter(sprinters,year <= 2004,olympics== 1)
```

b. Run the regressions

```
mod1 <- lm(time ~ year + women, data = sprinters_orig)
mod2 <- lm(time ~ year * women, data = sprinters_orig)
mod3 <- lm(time ~ year, data = filter(sprinters_orig, women == 1))
mod4 <- lm(time ~ year, data = filter(sprinters_orig, women == 0))
```

The model results are compiled and listed in the following table.

```
mtable1234 <- mtable('Model 1' = mod1,
                     'Model 2' = mod2,
                     'Model 3' = mod3,
                     'Model 4' = mod4,
                     summary.stats = c('R-squared','F','p','N'))
pander(mtable1234)
```

| | Model 1 | Model 2 | Model 3 | Model 4 |
|---|---|---|---|---|
| **(Intercept)** | 34.960*** | 31.826*** | 44.347*** | 31.826*** |
| | (1.965) | (2.129) | (4.284) | (1.680) |
| **year** | -0.013*** | -0.011*** | -0.017*** | -0.011*** |
| | (0.001) | (0.001) | (0.002) | (0.001) |
| **women** | 1.093*** | 12.521** | | |
| | (0.060) | (4.076) | | |
| **year x women** | | -0.006** | | |
| | | (0.002) | | |
| **R-squared** | 0.9 | 0.9 | 0.8 | 0.9 |
| **F** | 203.5 | 162.1 | 59.8 | 164.0 |
| **p** | 0.0 | 0.0 | 0.0 | 0.0 |
| **N** | 42 | 42 | 18 | 24 |

- Model 1: Winning times in the Olympic 100-meter dash would decrease by 0.01 seconds as the year increases 1. For the same year, compared to men, winning time of women would decrease by 1.09. The slop in this regession for women and men is the same: -0.01.
- Model 2: As the year increases by 1, winning times in the Olympic 100-meter dash would decrease by 0.01 seconds for men but decrease by (0.01+0.01)=0.02 seconds for women. For the same year, winning time of women would increase by (12.52-0.01*year) seconds compared to that of men. The slop in this regression for men is -0.01. But the slop for women is -0.02.
- Model 3: For female player, winning times in the Olympic 100-meter dash would decrease by 0.02 seconds as the year increases by 1. The slope of the regression  for women) is constant -0.02, which is as same as in Model 2, but different from that in Model 1.
- Model 4: For male player, winning times in the Olympic 100-meter dash would decrease by 0.01 seconds each year. The slope of the regression  for men) is constant -0.01, which is as same as in Model 1 and Model 2.

c. Plot the fitted values of these regressions against the original values.

```
#Model 1:
ggplot(new_Pre1, aes(x = year)) +
  geom_point(aes(y = time,colour=factor(women))) +
  geom_line(aes(y = .fitted,colour=factor(women)))+
  scale_color_manual("Gender",labels = c("men", "women"),values = c("blue", "red"))+
  ggtitle("Fig 10: fitted values vs. original values for Model 1")+
  theme_bw()+
  theme(text=element_text(size=10),
        axis.title=element_text(size=10),
        legend.position="right",
        legend.text=element_text(size=10))
```

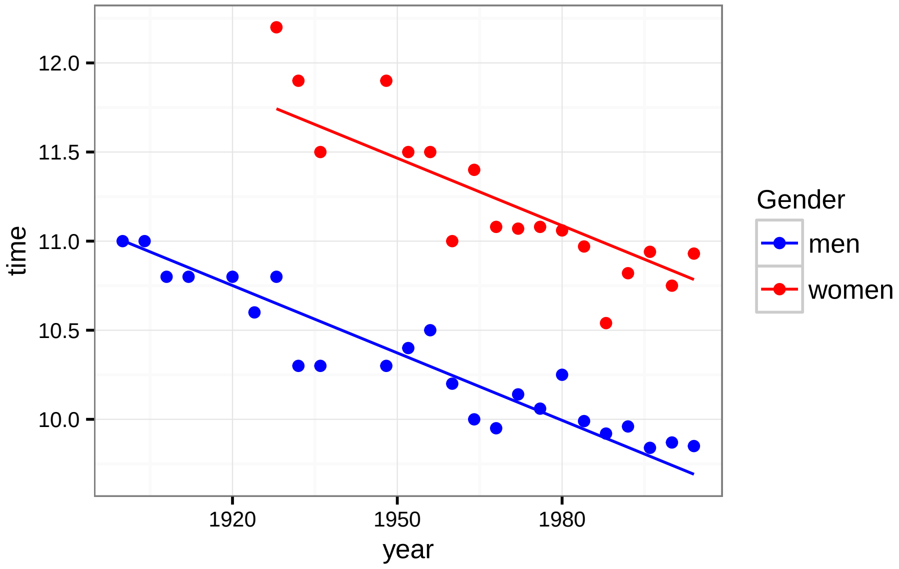
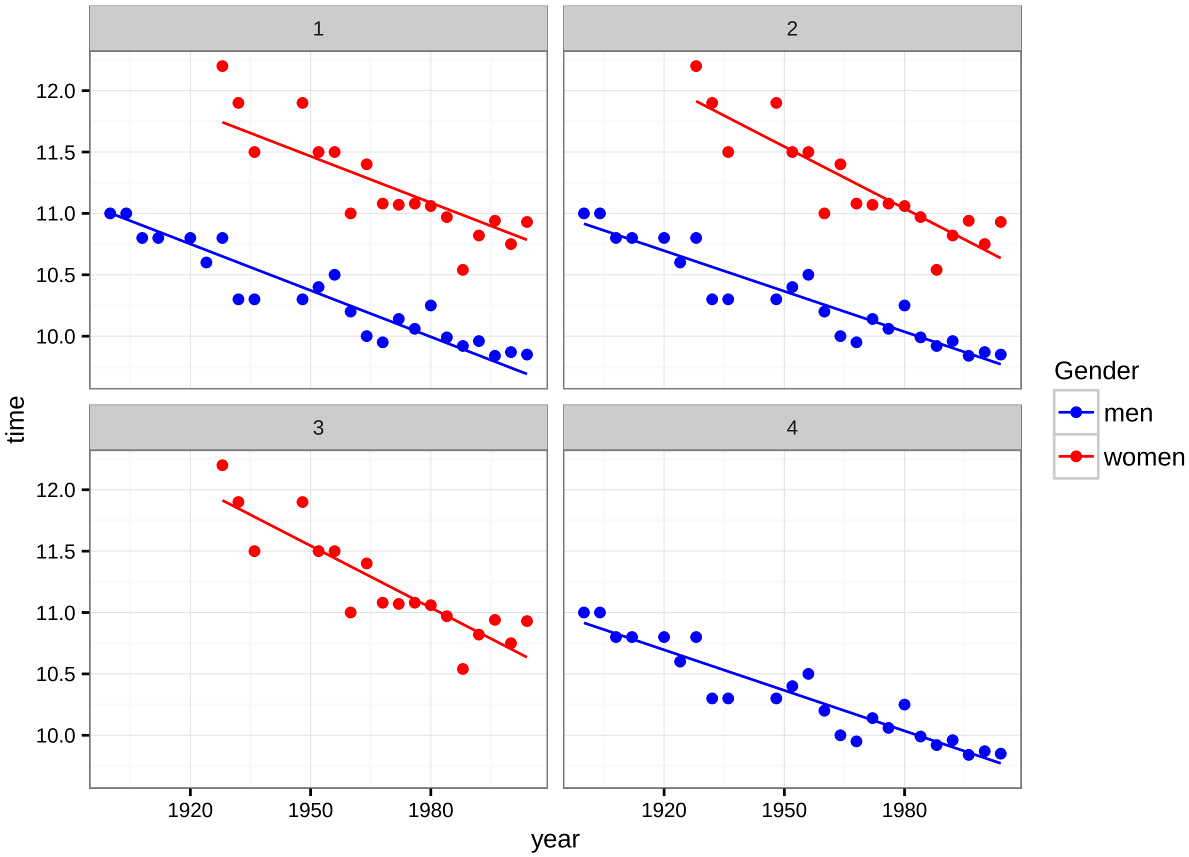


Fig 10: fitted values vs. original values for Model 1

```r
ggplot(models, aes(x = year)) +
  facet_wrap(~Model)+
  geom_point(aes(y = time,colour=factor(women))) +
  geom_line(aes(y = .fitted,colour=factor(women)))+
  scale_color_manual("Gender",labels = c("men", "women"),values = c("blue", "red"))+
  ggtitle("Fig 11: Fitted values vs. original values for each model")+
  theme_bw()+
  theme(text=element_text(size=10),
        axis.title=element_text(size=10),
        legend.position="right",
        legend.text=element_text(size=10))
```

Fig 11: Fitted values vs. original values for each model



d. Predict the times of men and women in the 2156 Olympics. The predicted winning times of 100 meter sprint for the 2156 Olympics based on above models are:

- Model 1: 7.77 seconds for male with a 95% confidence interval [7.36, 8.19]; 8.87 seconds for female, with a 95% confidence interval [8.48, 9.26].
- Model 2: 8.10 seconds for male with a 95% confidence interval [7.65, 8.55]; 8.08 seconds for female, with a 95% confidence interval [7.40, 8.75].
- Model 3: 8.08 seconds for male with a 95% confidence interval [7.21, 8.95]
- Model 4: 8.10 seconds for female, with a 95% confidence interval [7.73, 8.46].

```
new<-list(year=rep(2156,2),women=c(0,1))
#Model 1:
predict(mod1,newdata=new,interval = "confidence")
```

```
##    fit  lwr  upr
## 1 7.78 7.36 8.19
## 2 8.87 8.48 9.26
```

```
#Model 2:
predict(mod2,newdata=new,interval = "confidence")
```

```
##    fit  lwr  upr
## 1 8.10 7.65 8.55
## 2 8.08 7.40 8.75
```

```
#Model 3:
predict(mod3,list(year=2156,women=1),interval = "confidence")
```

```
##    fit  lwr  upr
## 1 8.08 7.21 8.95
```

```
#Model 4:
predict(mod4,list(year=2156,women=0),interval = "confidence")
```

```
##   fit  lwr  upr
## 1 8.1 7.73 8.46
```

I don't think these prediction are plausible. These models have fairly high R-squares and small standard errors for residuals, suggesting that they fits the data well. That is, they have good predictions for winning times of 100 meter sprint between 1900 and 2004. However, it's assumed that there's a stable decrease of wining time over time if using these models to predict the wining time in future Olympics. If this is the case, the wining time for females will drop below 0 seconds from the year of 2620, which certainly cannot be true. The problem is that, we assume a linear relationship between year and finishing time based on a relatively short period of time (1900 to 2004). Nonetheless, this assumption may not hold in a long period of time in the future. Besides, as this data were collected over time, a serial correlation may exist and lead to inefficient estimates of  and incorrect standard errors.

e.RMSE caculation

- The RMSE for the regression time ~ year * women (Olympics before 2004) is 0.16

```
## Model 2: time ~ year * women
RMSE_Before <- sqrt(mean((new_Pre2$time-new_Pre2$.fitted)^2))
print(RMSE_Before)
```

```
## [1] 0.162
```

- The RMSE for the predictions of the years after 2004 for both Olympics and World Championships is 0.23.

```
after2004<-filter(sprinters,year > 2004)
after2004$predictions<-predict(mod2,list(year=after2004$year,women=after2004$women))
RMSE_After <- sqrt(mean((after2004$predictions-after2004$time)^2))
print(RMSE_After)
```

```
## [1] 0.229
```

Compared to the estimation period, we have a much samller sample of data after 2004 (out-of-sample). Hence, it is possible that a model may do unusually well or badly in this validation or out-of-sample testing period merely by virtue of getting lucky or unlucky–e.g., by making the right guess about an unforeseeable upturn or downturn in the near future, or by being less sensitive than other models to an unusual event that happens at the start of the validation period.