Discussion 2

Rummerfield & Berman

10/13/2017

Plotting in R

The first data set, Nursing, from the Stat2Data textbook, includes characteristics obtained from a nursing home in New Mexico.

Here is a description of the variables in the data set:

Variables	Descriptions
Beds	Number of beds in the nursing home
InPatientDays	Annual medical in-patient days (in hundreds)
AllPatientDays	Annual total patient days (in hundreds)
PatientRevenue	Annual patient care revenue (in hundreds of dollars)
NurseSalaries	Annual nursing salaries (in hundreds of dollars)
FacilitiesExpend	Annual facilities expenditure (in hundreds of dollars)
Rural	1=rural or 0=non-rural

*Note: the `NurseSalaries` variable is the *total* salary per year for ALL nurses.

Let's say we are interested in if the annual salaries of all the nurses at a nursing facility are inversely related to the number of annual in-patient days. It seems plausible that nurses who were paid better may be more experienced and satisfied with their job and therefore could help patients recover more quickly. Clearly this is not the only variable that should be included. Think about what other variables might help explain the annual medical in-patient days.

```
## Load the data from Stat2Data
data("Nursing")
```

`ggplot2` is a package in R that allows R users to create elegant data visualizations. There are many commands that allow you to customize your plots including color, background, text size, labeling, etc. Nearly all plots can also be done using base R functions, but nothing really looks a nice as ggplot.

```
# Make sure you install the library: install.packages("ggplot2")
# Load the library
library(ggplot2)
```

Exploratory Data Analysis

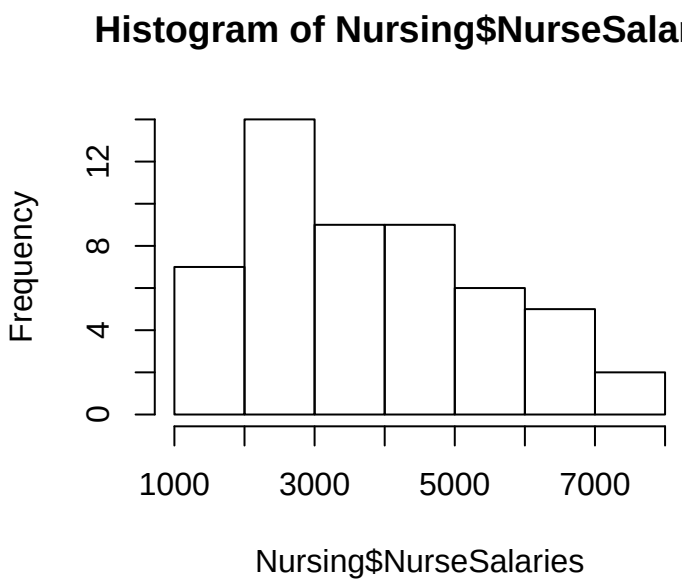
To get an idea of what a data set looks like, statisticians perform what we call an *Exploratory Data Analysis* (EDA).

```
#### Summary statistics
summary(Nursing[, c(2, 5)])
```

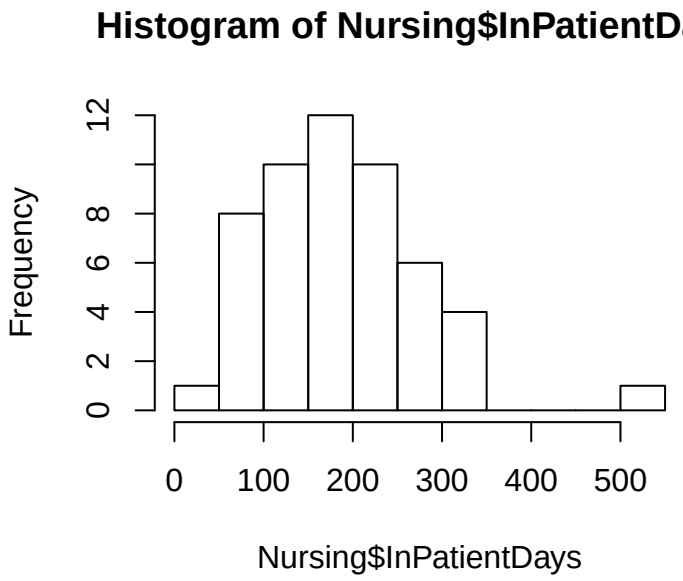
```
##  InPatientDays   NurseSalaries
##  Min.   : 48.0   Min.   :1288
##  1st Qu.:125.2   1st Qu.:2336
##  Median :164.5   Median :3696
##  Mean   :183.9   Mean   :3813
##  3rd Qu.:229.0   3rd Qu.:4840
```

```
##  Max.   :514.0   Max.   :7489
```

```
#### Check for outliers in x and y
# Histogram using base R libraries

hist(Nursing$NurseSalaries)
```

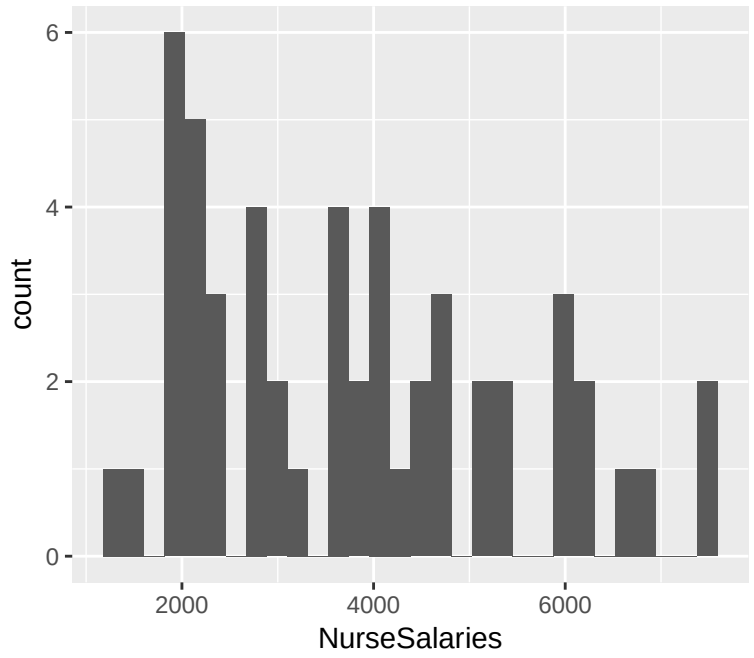



```
hist(Nursing$InPatientDays) # Looks like there is one extreme outlier
```

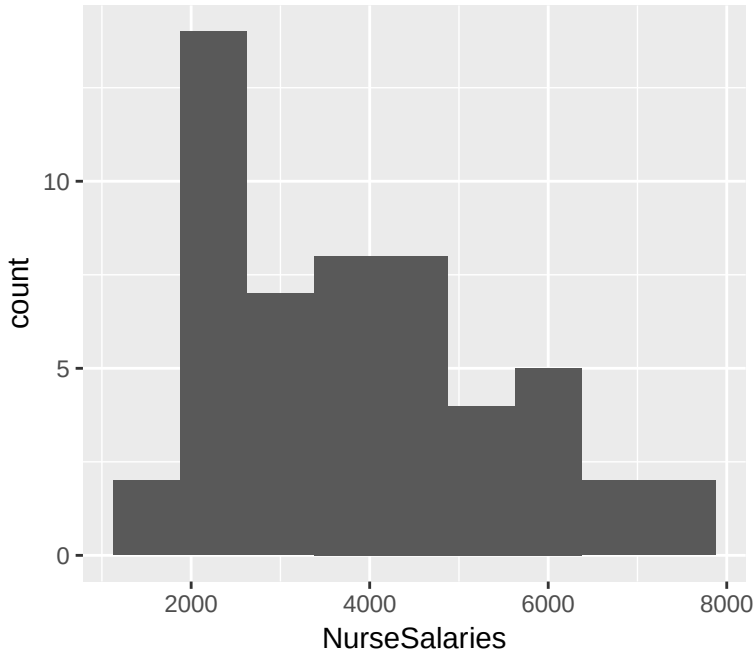



```
# Histogram using ggplot2
ggplot(Nursing) + geom_histogram(aes(x = NurseSalaries)) # binwidth is weird
```

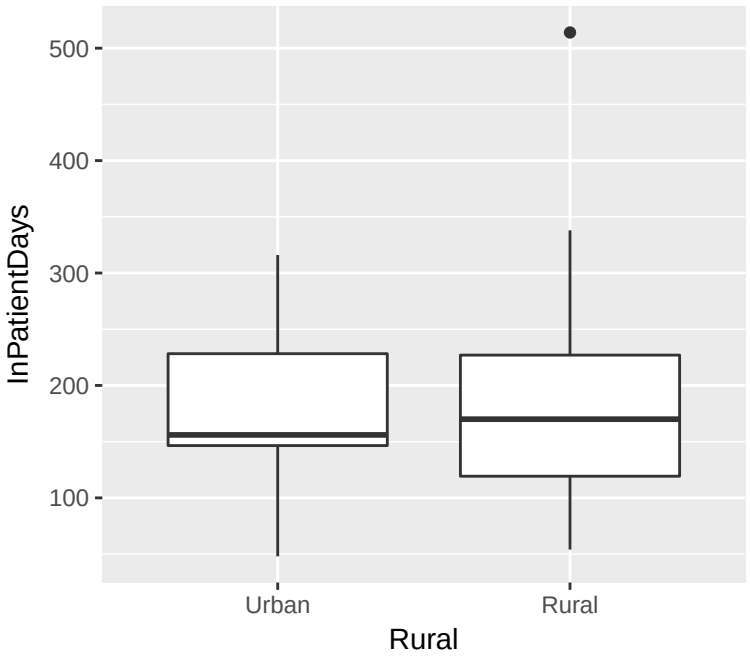
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot(Nursing) + geom_histogram(aes(x = NurseSalaries), binwidth = 750) # better binwidth
```



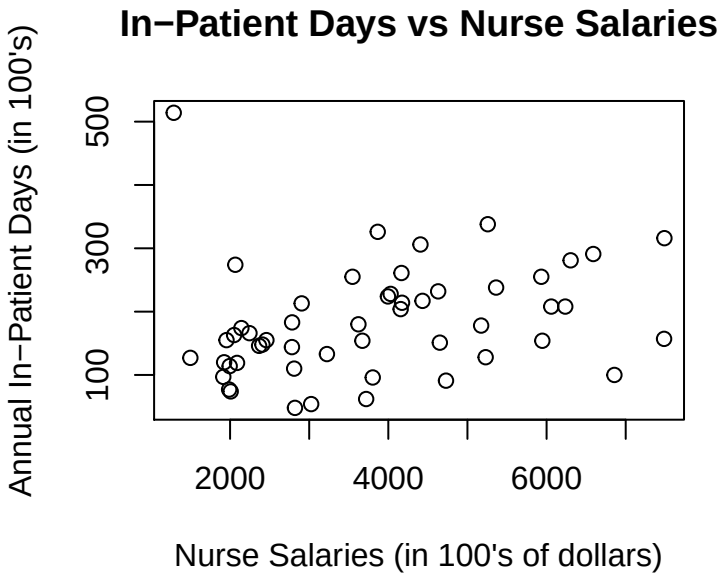

```
#### How does the number of in-patient days change depending on the environment?
## Use a boxplot!

# First, we need to make "Rural" a factor
Nursing$Rural <- factor(Nursing$Rural, levels = c(0,1), labels = c("Urban", "Rural"))
ggplot(Nursing) + geom_boxplot(aes(x = Rural, y = InPatientDays))
```



Prediction

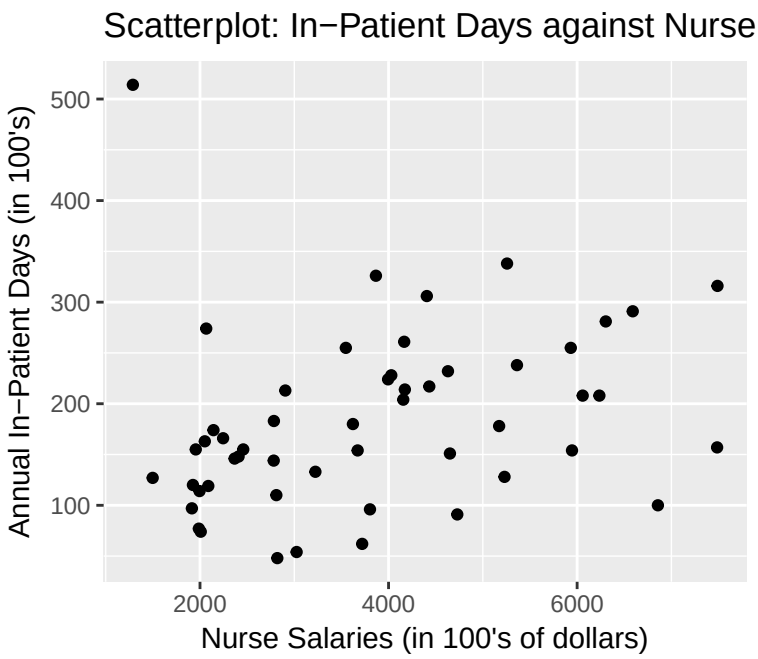
Now to the modeling. To determine if it makes sense to model the relationship between the number of annual in-patient days and nurse salaries linearly, we should use a scatterplot.

```
# (Well labeled) scatterplot using base R functions
plot(InPatientDays ~ NurseSalaries, data = Nursing,
     main = "In-Patient Days vs Nurse Salaries",
     xlab = "Nurse Salaries (in 100's of dollars)",
     ylab = "Annual In-Patient Days (in 100's)")
```



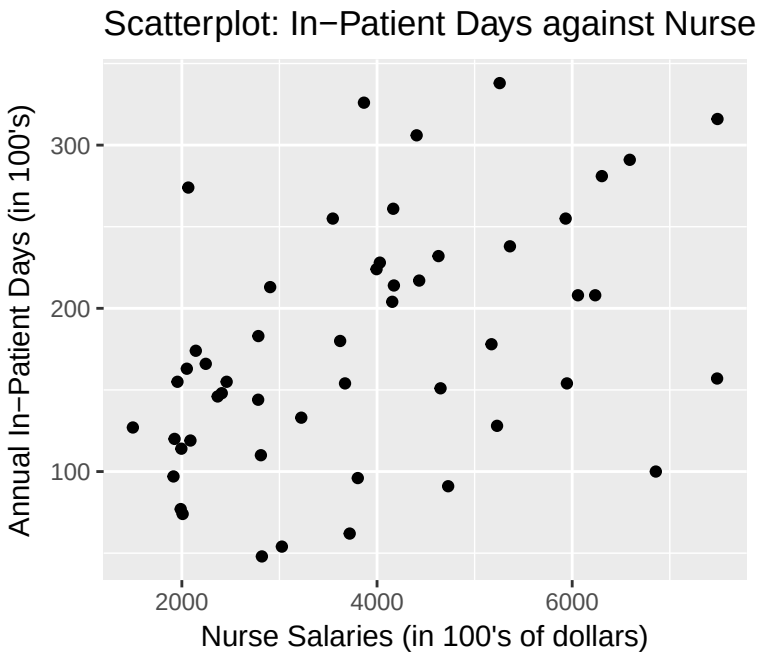

```
# (Well labeled) scatterplot using ggplot2
ggplot(Nursing) + geom_point(aes(x = NurseSalaries, y = InPatientDays)) +
  ggtitle("Scatterplot: In-Patient Days against Nurse Salaries") +
```

```
  xlab("Nurse Salaries (in 100's of dollars)") +
  ylab("Annual In-Patient Days (in 100's)")
```



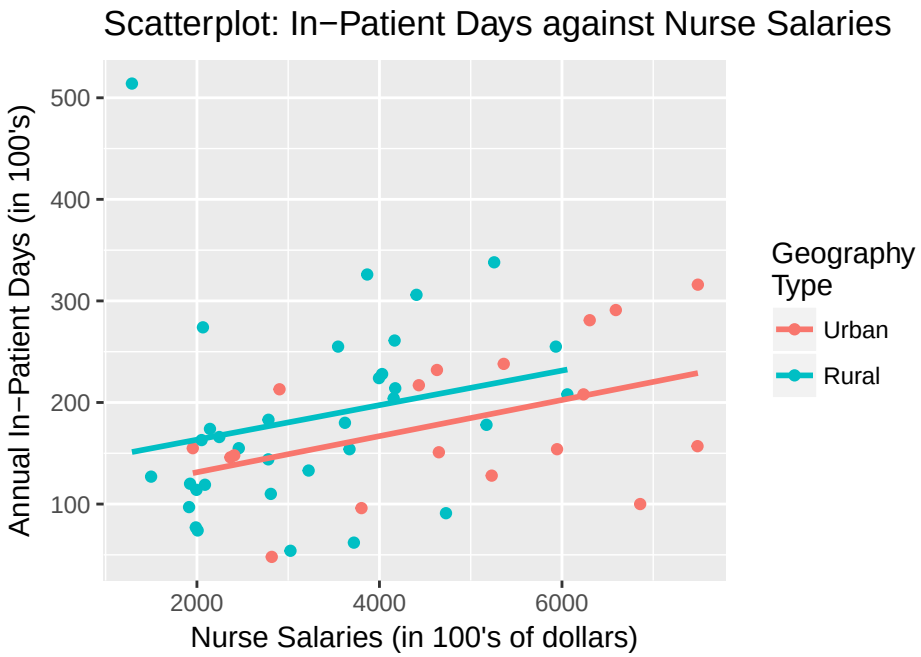
There is a very noticable outlier. Make a new scatterplot with this value removed.

```
ggplot(Nursing[!(Nursing$InPatientDays == 514), ]) + geom_point(aes(x = NurseSalaries, y = InPatientDays
  ggtitle("Scatterplot: In-Patient Days against Nurse Salaries") +
  xlab("Nurse Salaries (in 100's of dollars)") +
  ylab("Annual In-Patient Days (in 100's)")
```




```
## Let's see how this relationship changes by geography
ggplot(Nursing, aes(x = NurseSalaries, y = InPatientDays, color = Rural)) +
  geom_point() + geom_smooth(method = "lm", formula = y ~ x, se = F) +
```

```
  ggtitle("Scatterplot: In-Patient Days against Nurse Salaries") +
  xlab("Nurse Salaries (in 100's of dollars)") +
  ylab("Annual In-Patient Days (in 100's)") +
  scale_color_discrete("Geography \nType")
```



We can see that the linear relationship doesn't really seem to change much whether or not the hospital is in an urban or rural environment.

Now predict the annual number of in-patient beds given that the nurses make $195,500 annual at a nursing facility. Compare this value to the observed recorded number of in-patient bed and calculate the residual.

```
mod1 <- lm(InPatientDays ~ NurseSalaries, data = Nursing)
summary(mod1)
```

```
##
## Call:
## lm(formula = InPatientDays ~ NurseSalaries, data = Nursing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -124.44  -48.80   -6.98   38.59  363.79
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.330e+02  2.978e+01   4.467 4.54e-05 ***
## NurseSalaries 1.333e-02  7.173e-03   1.858   0.0691 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85.01 on 50 degrees of freedom
## Multiple R-squared:  0.06458,    Adjusted R-squared:  0.04588
## F-statistic: 3.452 on 1 and 50 DF,  p-value: 0.06906
```

```
mod1.a <- lm(InPatientDays ~ NurseSalaries, data = Nursing[!(Nursing$InPatientDays == 514), ])
summary(mod1.a)
```

```
##
## Call:
## lm(formula = InPatientDays ~ NurseSalaries, data = Nursing[!(Nursing$InPatientDays ==
##     514), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -138.242  -41.610    0.139   37.271  148.533
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   98.911853  24.241748   4.080 0.000165 ***
## NurseSalaries  0.020319   0.005788   3.511 0.000969 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 66.98 on 49 degrees of freedom
## Multiple R-squared:  0.201,  Adjusted R-squared:  0.1847
## F-statistic: 12.33 on 1 and 49 DF,  p-value: 0.0009693
```

```
## Predict number of in-patient beds without removing the outlier
summary(mod1)$coef[1] + 1955*summary(mod1)$coef[2]
```

```
## [1] 159.1045
```

```
Nursing$InPatientDays[Nursing$NurseSalaries == 1955]
```

```
## [1] 155
```

```
# Residual = observed - predicted
```

Transformations and Conditions for Simple Linear Regression

The next data set we will look at comes from the United Nations (UN) about the national health, welfare, and education for 210 countries. This time we are interested in how the number of children per woman in Africa is influenced by the gross domestic product (gdp). Our goal is to perform a hypothesis test to determine if there is a linear relation between these two variables. Recall the steps of a hypothesis test: (1) Establish Hypotheses, (2) Check condition, (3) Calculate a p-value, (4) Make a decision, (5) Conclusion within context.

Step 1: Establish hypotheses:

We want to test the linear model, $\widehat{\text{fertility}} = \beta_0 + \beta_1 \times \text{GDP}$, versus, the constant model, $\widehat{\text{fertility}} = \beta_0$.

Step 2: Check conditions of linear regression:

In order to perform simple linear regression we need to check three (four) assumptions.

It may help to use the acronym **L.I.N.E.**

Linear relationship

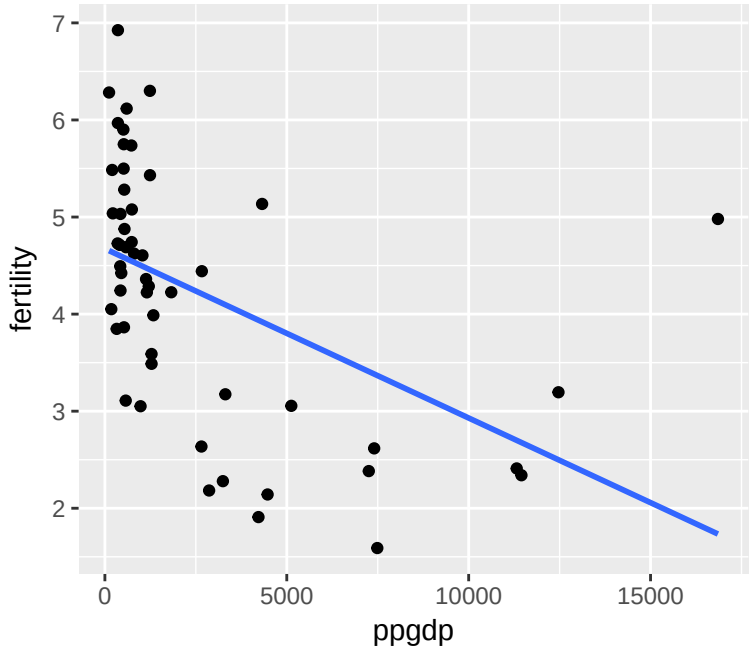
Independent observations

Normality of the residuals

Equal (constant) variance

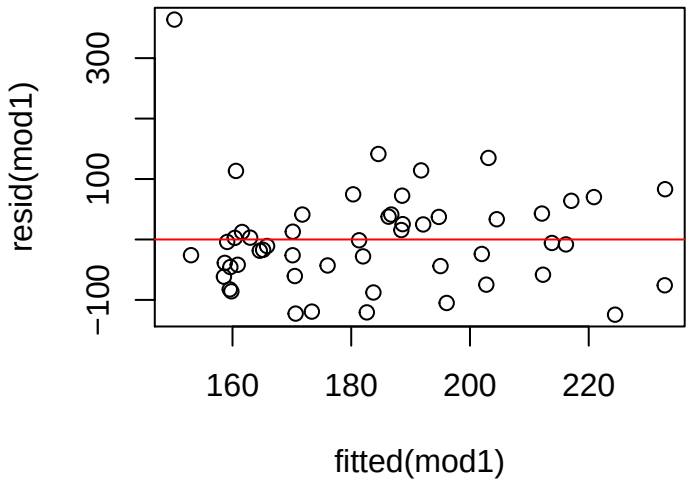
1. Linearity:

```
## Load in the data
library(alr4)
data(UN11)

ggplot(UN11[UN11$region == "Africa", ], aes(x = ppgdp, y = fertility)) + geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE)
```



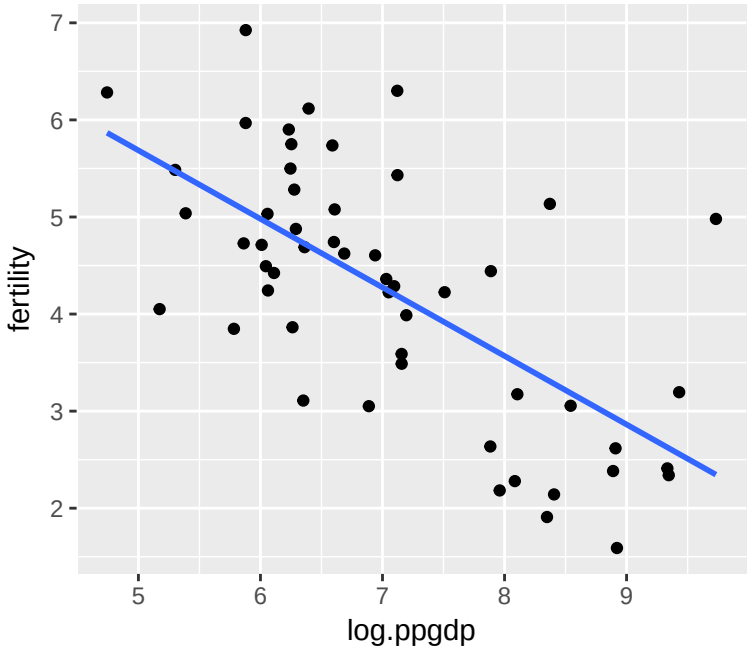
This definitely does not look linear. Take a look at the residuals vs. fitted (or x) plot to try and figure out an appropriate transformation

```
plot(resid(mod1) ~ fitted(mod1))
abline(h = 0, col = "red")
```



This plot doesn't look too bad so let's go back and consider the data we are working with. You'll notice from the scatterplot that a lare proportion of the observations are below $2,500,000, but there are some countries with vary large per capita gross domestic product. You see this kind of data a lot in economics and it is common to use a log transformation to pull the extreme points back.

```
# Create a new variable: log.ppgdp
UN11$log.ppgdp <- log(UN11$ppgdp)

ggplot(UN11[UN11$region == "Africa", ], aes(x = log.ppgdp, y = fertility)) + geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE)
```



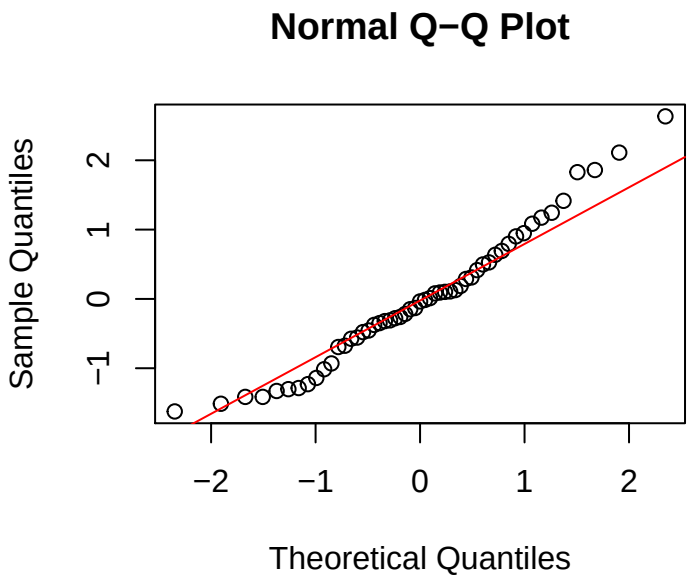
2. Independence:

Topic is too advanced for this class.

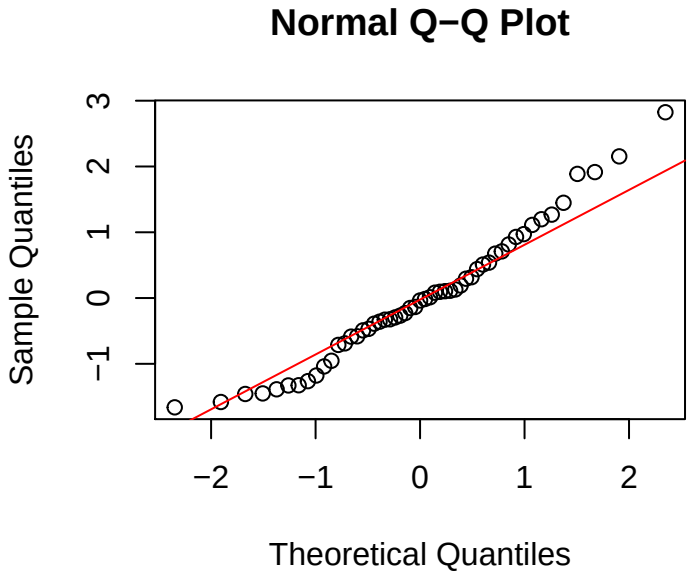
3. Normality of the residuals:

To check if the residuals are normal, we use quantile-quantile (qq)-plots.

```
## Obtain the residuals
mod2 <- lm( fertility ~ log.ppgdp, data = UN11[UN11$region == "Africa", ])

## qq-plot
qqnorm(resid(mod2))
qqline(resid(mod2), col = "red")
```



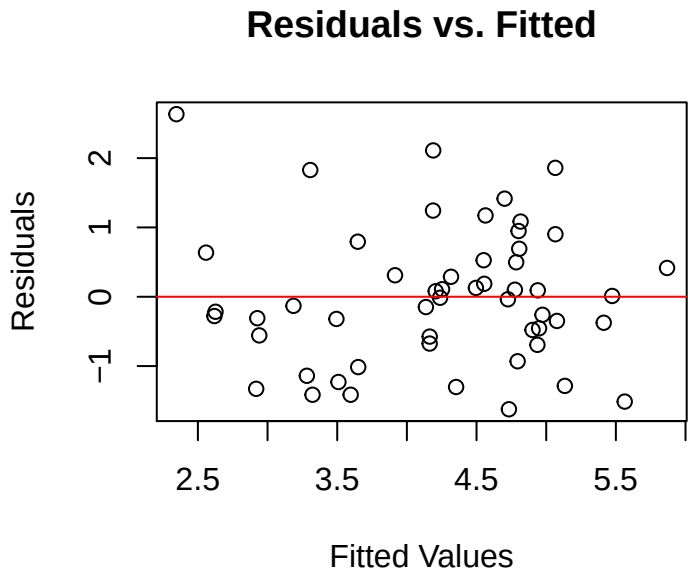

```
## qq-plot with standardized residuals
qqnorm(rstandard(mod2))
qqline(rstandard(mod2), col = "red")
```



4. Equal (constant) variance:

Finally, to check equal (constant) variance we use a residuals vs. fitted plot.

```
plot(resid(mod2) ~ fitted(mod2), main = "Residuals vs. Fitted",
     xlab = "Fitted Values", ylab = "Residuals")
abline(h = 0, col = "red")
```



After assuring all of the conditions are met, you can calculate the test statistic.

Step 3: Calculation p-value

```
summary(mod2)
```

```
##
## Call:
## lm(formula = fertility ~ log.ppgdp, data = UN11[UN11$region ==
##     "Africa", ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.62351 -0.57484 -0.03616  0.52696  2.63431
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.2158     0.8064  11.428 1.12e-15 ***
## log.ppgdp    -0.7059     0.1127  -6.265 7.87e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9896 on 51 degrees of freedom
## Multiple R-squared:  0.4349, Adjusted R-squared:  0.4238
## F-statistic: 39.25 on 1 and 51 DF,  p-value: 7.872e-08
```

Step 4: Make a decision

The p-value for the test, $H_0: \beta_1 = 0$ versus $H_a: \beta_1 \neq 0$, is very small, we should reject the null hypothesis.

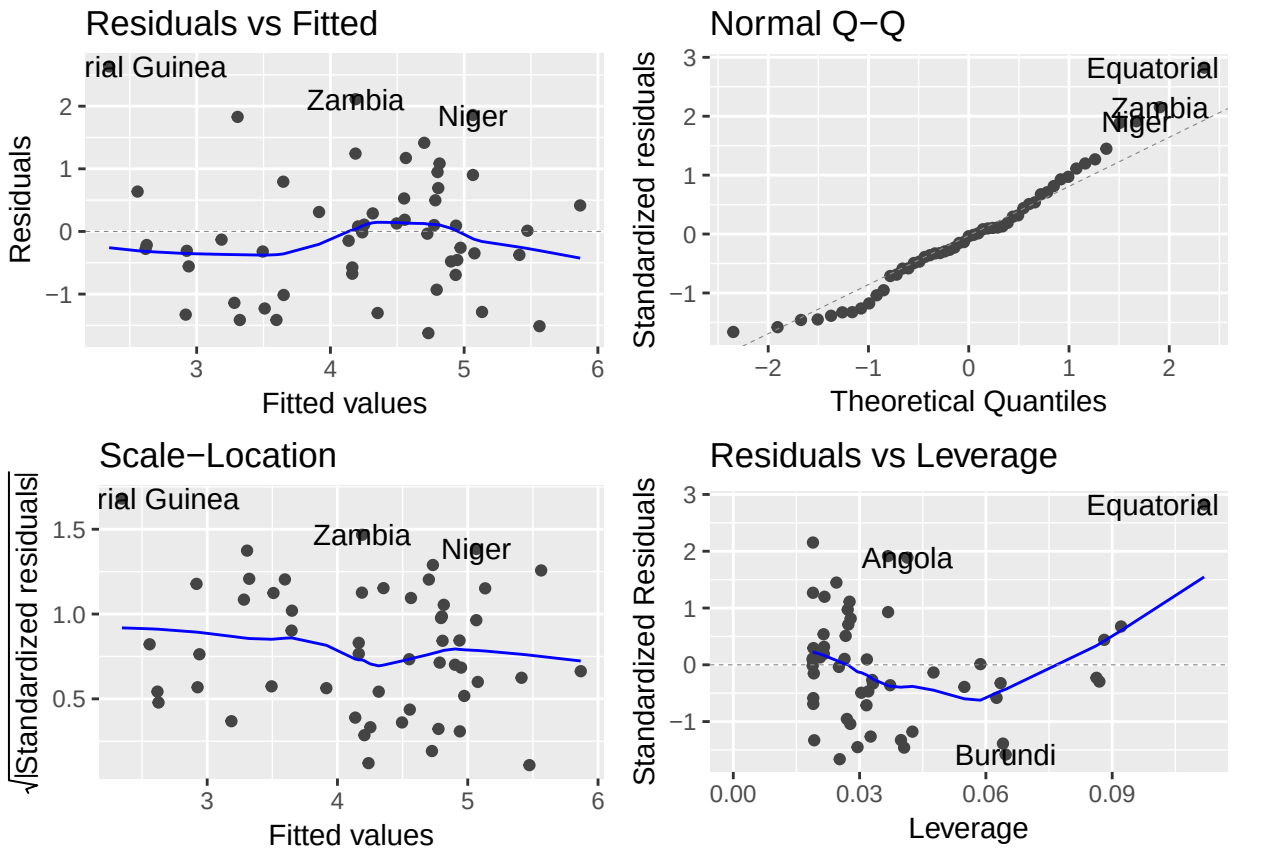
Step 5: Conclusion in context.

Based on this data, we would conclude that by including the log of per capita GDP we better predict the fertility rate among African countries than we would by using the average fertility rate among all the Afrian countries.

More on plotting with ggplot2

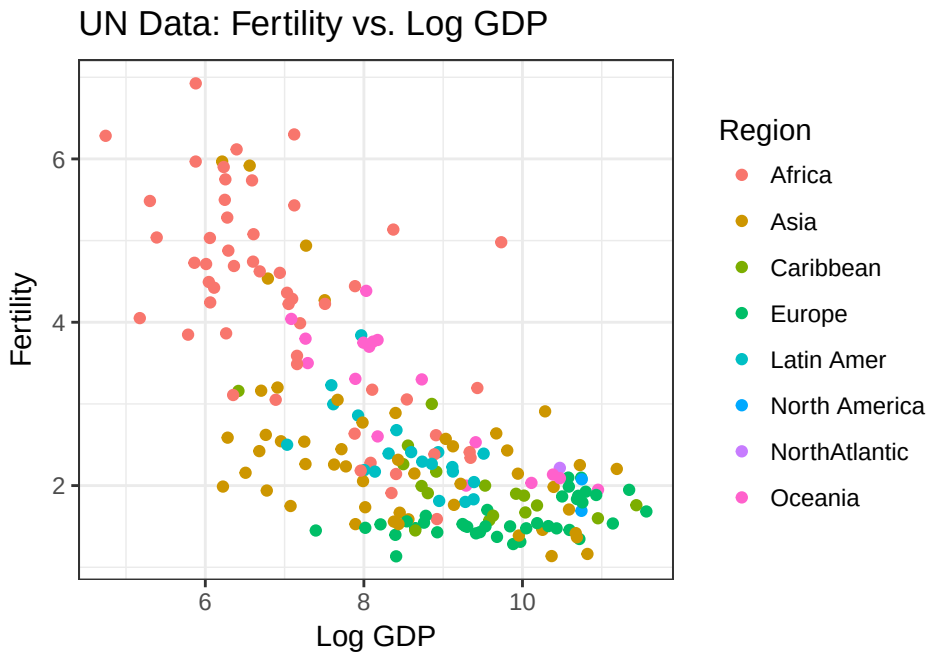
First, to get a grid of useful plots for linear modeling, use the command `autoplot` in the `ggfortify` library.

```
library(ggfortify)
autoplot(mod2)
```



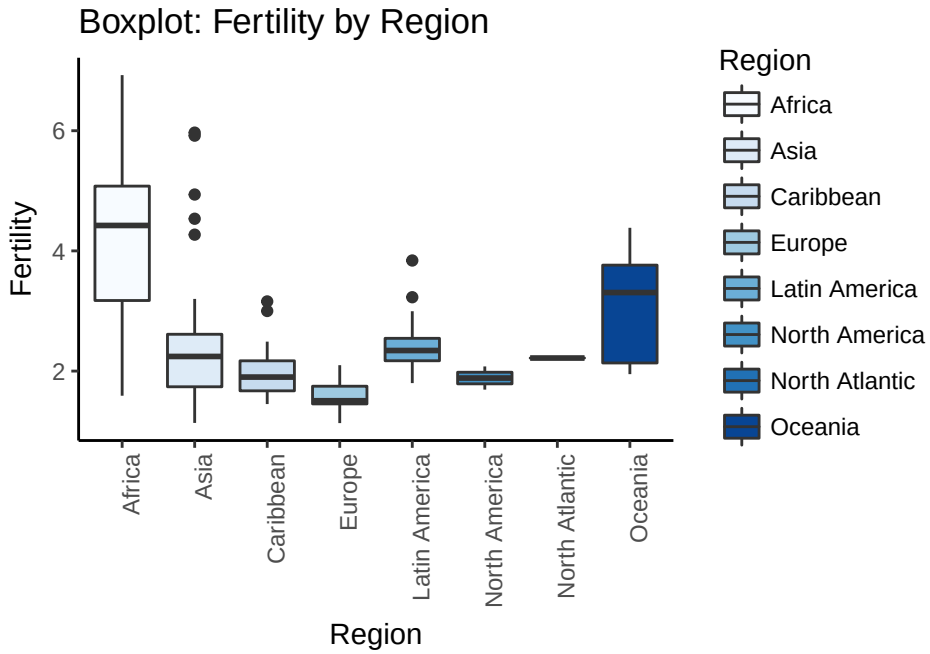
There is whole slew of customizations you can do in ggplot. All you have to do to make small changes is add a '+'.

```
ggplot(UN11, aes(x = log.ppgdp, y = fertility, color = region)) +
  geom_point() +
  labs(x = "Log GDP", y = "Fertility", title = "UN Data: Fertility vs. Log GDP") +
  theme_bw() + scale_color_discrete(name = "Region")
```



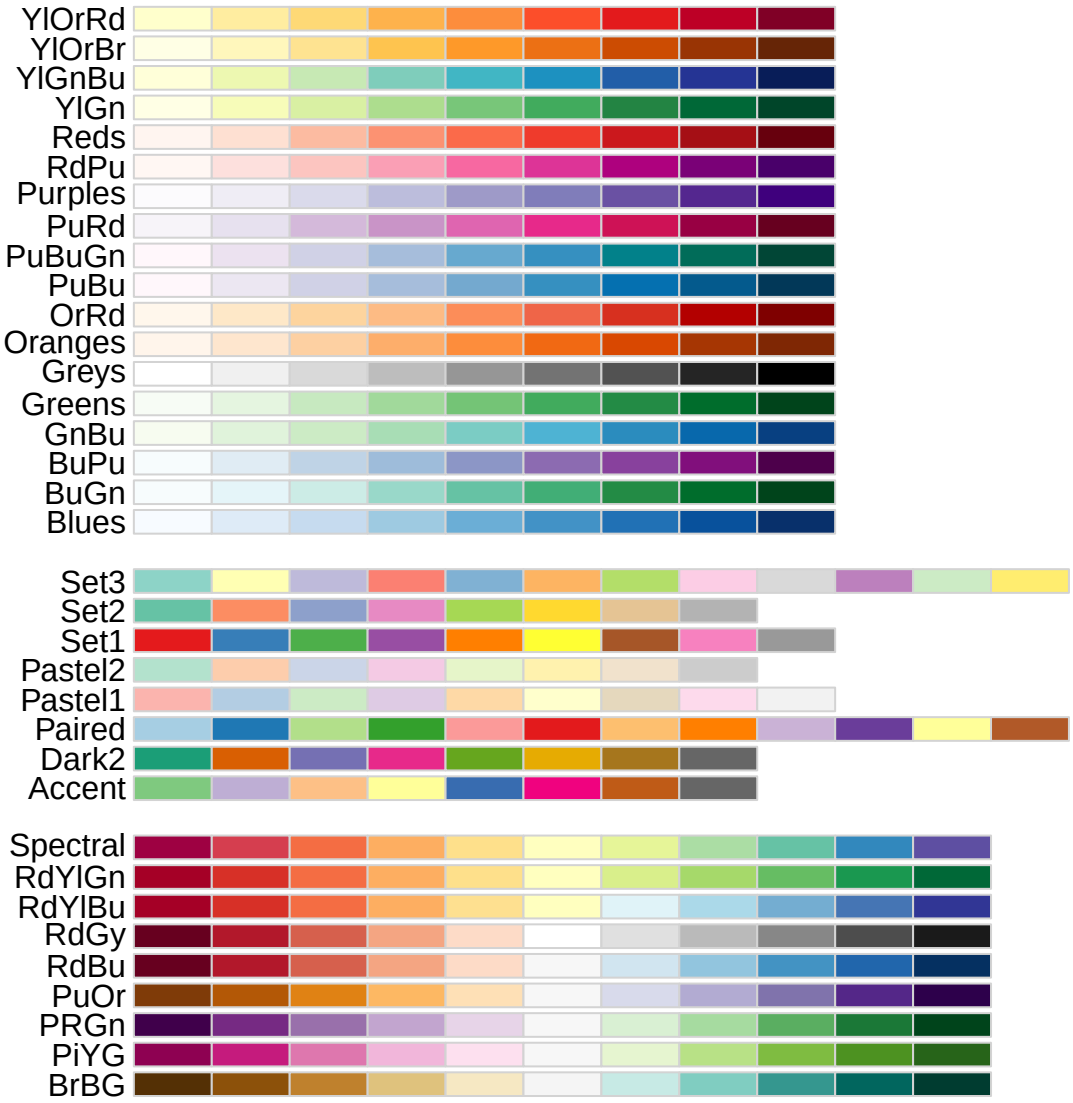
- `theme_bw()`: instead of a grey background with grid lines you get a white background with grid lines
- `scale_color_discrete()`: Amongst much else, this allows you to change the title of the legend

```
# Let's rename the factor levels to full name of the region
levels(UN11$region) = c("Africa", "Asia", "Caribbean", "Europe", "Latin America", "North America", "Nor

# Remember: you need to install the package the first time you use it
library(RColorBrewer)

ggplot(UN11, aes(x = region, y = fertility, fill = region)) +
  geom_boxplot() + labs(title = "Boxplot: Fertility by Region",
                        x = "Region", y = "Fertility", fill = "Region") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_brewer(palette = "Blues")
```



- `fill`: fill in the boxplots with a different color for each factor
- `color`: changed the outline color of the boxplots
- `theme(axis.text.x = element_text(angle = 90, hjust = 1))`: since the region names were so long they overlapped on the x axis- this code rotates them 90 degrees
- `scale_fill_brewer(palette = "Blues")`: you can change the color scheme using any one of the palettes in the `RColorBrewer` package- the following code allows you to see all of the palettes they have

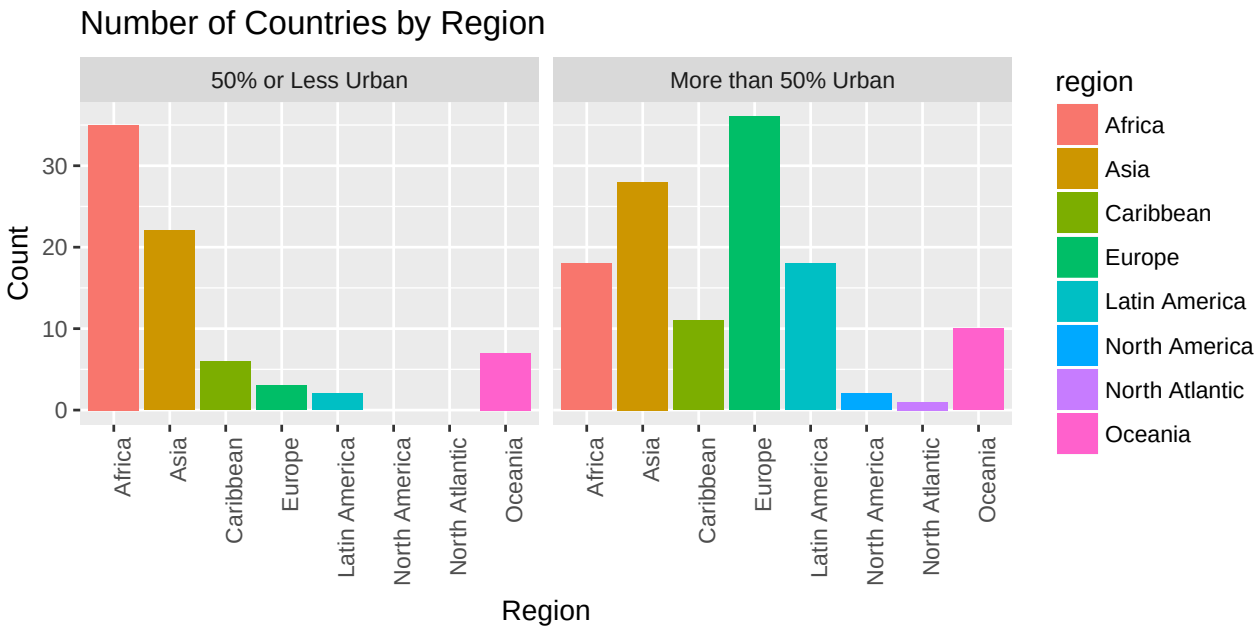
```
display.brewer.all()
```




```
summary(UN11$pctUrban)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   39.00   59.00   57.93   75.00  100.00
```

```
# Make pctUrban a factor with 2 levels: <= 50% or > 50%
UN11$pctUrban_factor <- cut(UN11$pctUrban, breaks = c(-1, 50, 101))
# Rename the levels so it looks nice in the plot
levels(UN11$pctUrban_factor) <- c("50% or Less Urban",
                                  "More than 50% Urban")

ggplot(UN11, aes(x = region, fill = region)) + geom_bar() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(x = "Region", y = "Count", title = "Number of Countries by Region") +
  facet_wrap(~pctUrban_factor)
```



- `facet_wrap( ~ )`: by putting one of the factors here you can make separate plots side by side for each level of the factor (you need the $\sim$ before the factor)