# Week 3 Notebook

## Contents

Firstly, importing the libraries used in this notebook

```
library('tidyverse')
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library('cowplot')
```

# 1 Part 1

## 1.1 Data

The file wb_warwick.csv contains fictitious well-being data of 30 students from Warwick. In total the data set has 4 columns/variables.

```
dw <- read_csv('wb_warwick.csv')
```

```
## Rows: 30 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (4): wb, ig, cooked, rolf

##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
glimpse(dw)
```

```
## Rows: 30
## Columns: 4
## $ wb     <dbl> 5, 7, 4, 7, 7, 9, 8, 6, 8, 5, 7, 4, 10, 2, 5, 6, 4, 6, 4, 1, 6,~
## $ ig     <dbl> 121, 90, 128, 0, 127, 44, 83, 119, 9, 103, 98, 118, 0, 76, 41, ~
## $ cooked <dbl> 5, 7, 6, 7, 3, 7, 7, 7, 7, 7, 6, 3, 7, 2, 4, 6, 5, 5, 2, 4, 7, ~
## $ rolf   <dbl> 2, 0, 1, 1, 1, 1, 0, 0, 1, 0, 2, 1, 0, 1, 1, 2, 1, 1, 1, 1, 0, ~
```

The variables in the data are:

- `wb`: Subjective well-being score on a scale from 1 to 10.
- `ig`: Number of Instagram followers.
- `cooked`: Times the student has cooked instead of eating a ready meal or fast food in the last week.
- `rolf`: Number of times the student has seen Rolf (the campus cat) in the last week.

The main research question for the data is, which of the 3 independent variables (ig, cooked, rolf) predict the subjective well-being score.

## 1.2 Task 1

Setting up individual simple linear regressions for each of the independent variables predicting subjective well-being:

We estimate 3 models:

- Model 1:
$$wb = \beta_0 + \beta_1 ig$$
- Model 2:
$$wb = \beta_0 + \beta_1 cooked$$
- Model 3:
$$wb = \beta_0 + \beta_1 rolf$$

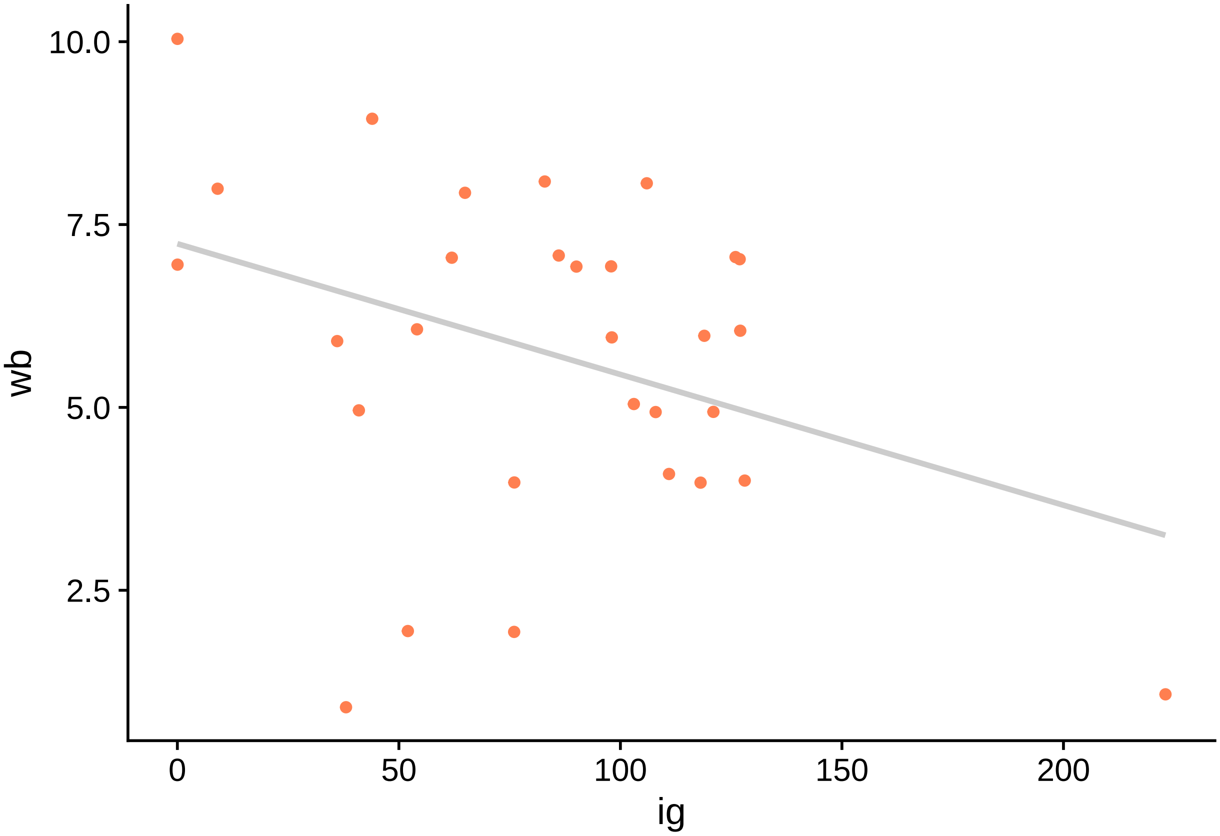
First, for Model 1

```
fit1 <- lm(wb ~ ig, dw)
summary(fit1)
```

```
##
## Call:
## lm(formula = wb ~ ig, data = dw)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5583 -1.0841  0.2194  1.4781  2.7627
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.237330   0.812235    8.91 1.15e-09 ***
## ig          -0.017869   0.008467   -2.11   0.0439 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.135 on 28 degrees of freedom
## Multiple R-squared:  0.1372, Adjusted R-squared:  0.1064
## F-statistic: 4.454 on 1 and 28 DF,  p-value: 0.04388
```

```
p1 <- ggplot(dw, mapping=aes(x=ig,y=wb)) +
  geom_jitter(height=0.1,width=0.1, color='coral') + geom_smooth(method='lm',color='grey80',se=FALSE) +
  theme_cowplot()
p1
```



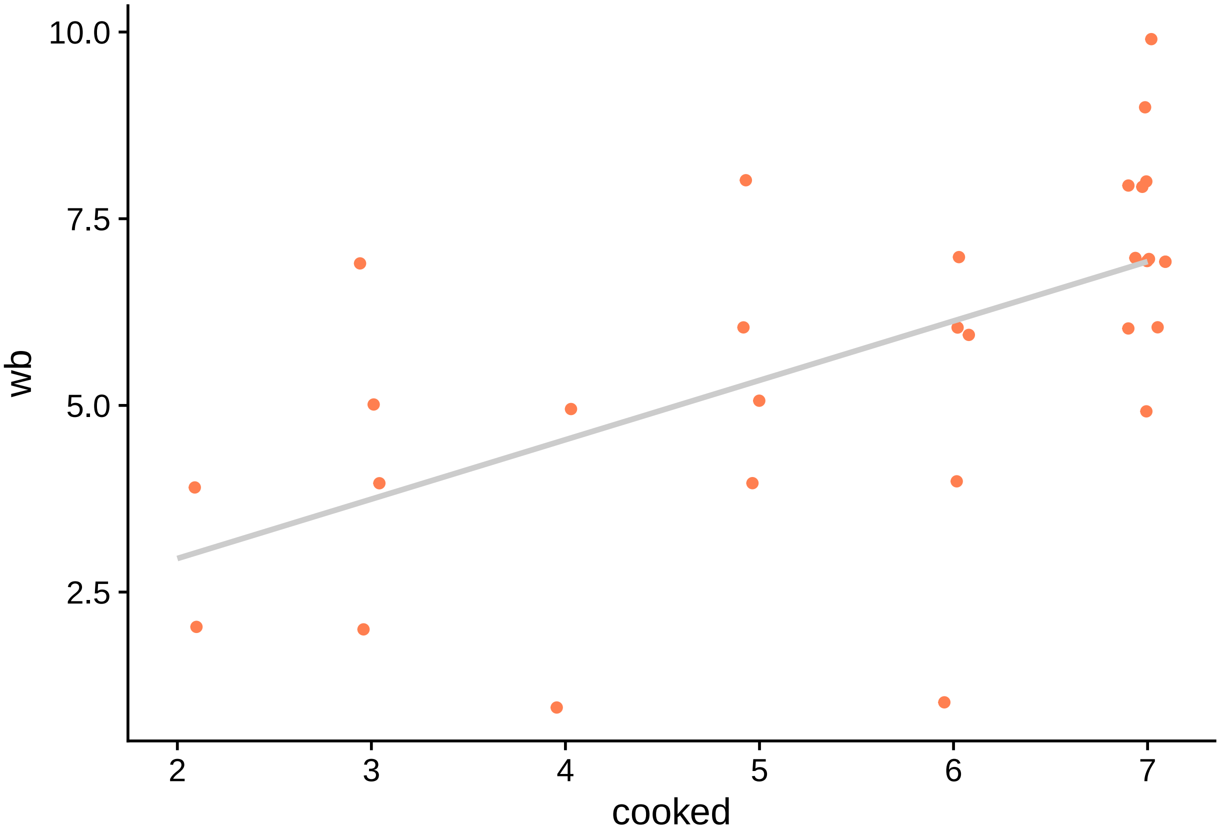
Note the outlier at `ig` $> 200$.

Next, for Model 2

```
fit2 <- lm(wb ~ cooked, dw)
summary(fit2)
```

```
##
## Call:
## lm(formula = wb ~ cooked, data = dw)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1310 -0.9263  0.0737  1.0678  3.2550
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3591     1.1387   1.193   0.2427
## cooked        0.7953     0.1979   4.018   0.0004 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.83 on 28 degrees of freedom
## Multiple R-squared:  0.3657, Adjusted R-squared:  0.3431
## F-statistic: 16.15 on 1 and 28 DF,  p-value: 0.0004001
```

```
p2 <- ggplot(dw, mapping=aes(x=cooked,y=wb)) +
  geom_jitter(height=0.1,width=0.1, color='coral') + geom_smooth(method='lm',color='grey80',se=FALSE) +
  theme_cowplot()
p2
```



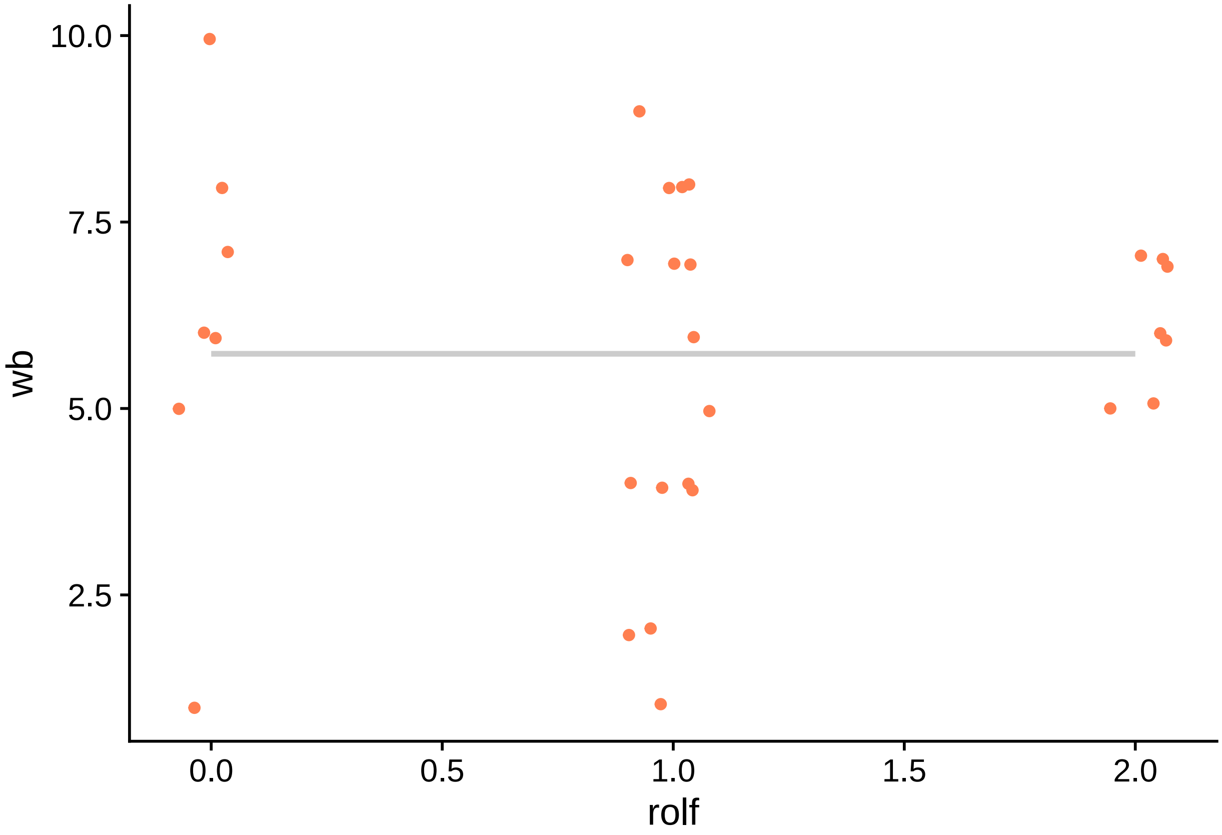
Note that most data points for `cooked` are at the extreme end at 7, it is highly right-skewed.

Finally, for Model 3

```
fit3 <- lm(wb ~ rolf, dw)
summary(fit3)
```

```
##
## Call:
## lm(formula = wb ~ rolf, data = dw)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.7333 -1.4833  0.2667  1.2667  4.2667
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.733e+00  7.438e-01   7.708 2.14e-08 ***
## rolf        -5.341e-16  6.142e-01   0.000        1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.298 on 28 degrees of freedom
## Multiple R-squared:  8.429e-31,  Adjusted R-squared:  -0.03571
## F-statistic: 2.36e-29 on 1 and 28 DF,  p-value: 1
```

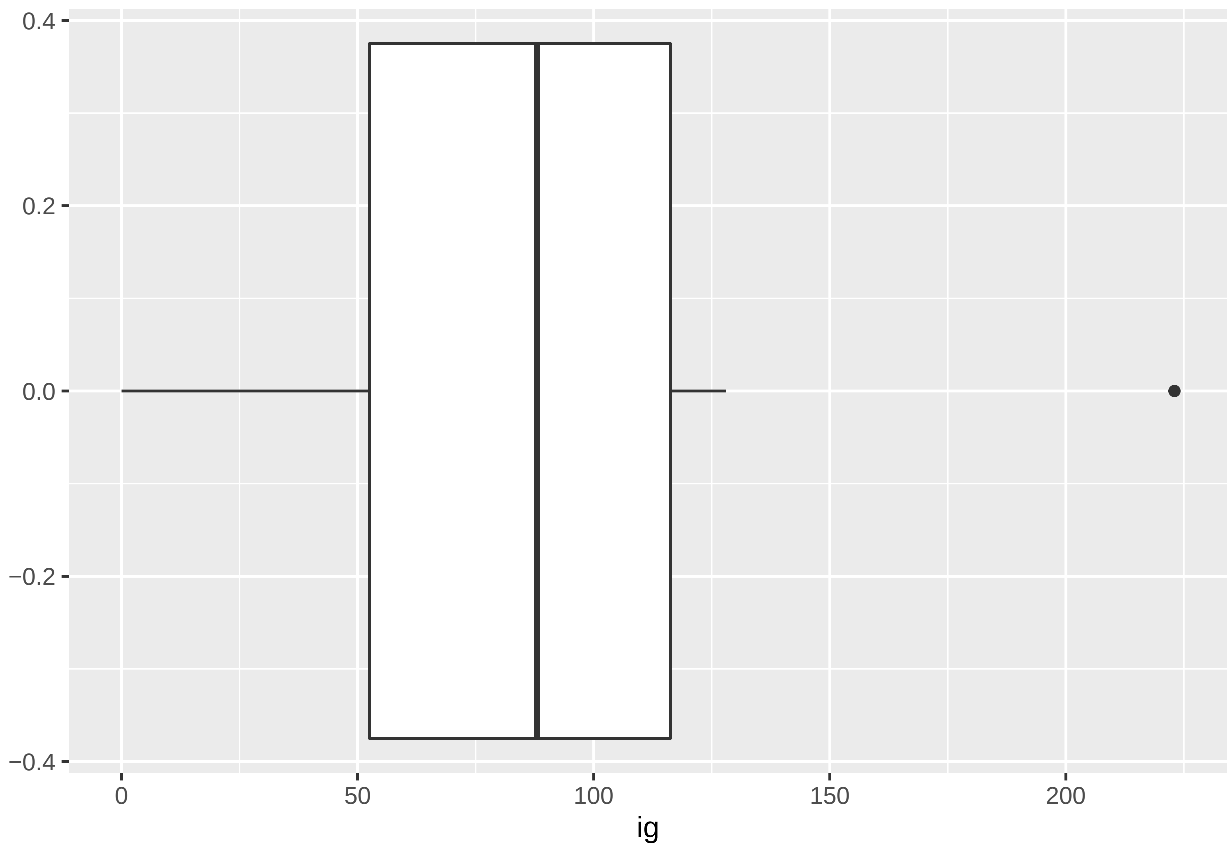
```
p3 <- ggplot(dw, mapping=aes(x=rolf,y=wb)) +
  geom_jitter(height=0.1,width=0.1, color='coral') + geom_smooth(method='lm',color='grey80',se=FALSE) +
  theme_cowplot()
p3
```



**Do any of the plots show a suspicious pattern, which could be problematic when applying linear regression?**

The `ig` plot shows us that there may exist an outlier at `ig` $> 200$.

```
ggplot(dw,mapping=aes(x=ig)) + geom_boxplot()
```



Thus, we should rerun the regressions after removing the outlier. Further, the data for `cooked` is highly right-skewed and may not provide accurate results.

Removing the outlier,

```
dw2 <- dw %>% filter(ig<200)
```

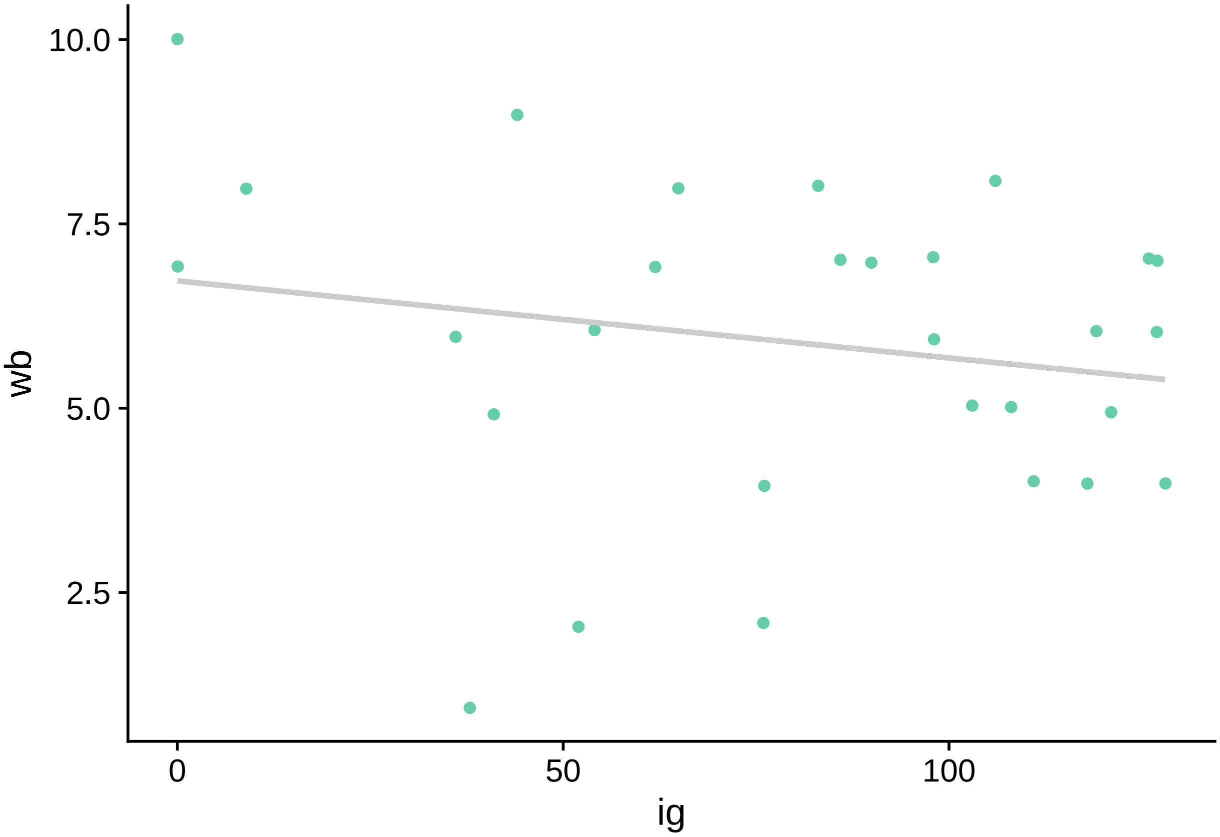
And regressing Model 1 again,

```
fit1_n <- lm(wb ~ ig, dw2)
summary(fit1_n)
```

```
##
## Call:
## lm(formula = wb ~ ig, data = dw2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3290 -1.2977  0.2981  1.3678  3.2738
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.72622    0.89155   7.544 4.08e-08 ***
## ig          -0.01045    0.01009  -1.036     0.31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.108 on 27 degrees of freedom
## Multiple R-squared:  0.03821,    Adjusted R-squared:  0.00259
## F-statistic: 1.073 on 1 and 27 DF,  p-value: 0.3095
```

The variable is no longer statistically significant, suggesting `ig` might have no effect on well-being

```
p1_n <- ggplot(dw2, mapping=aes(x=ig,y=wb)) +
  geom_jitter(height=0.1,width=0.1, color='aquamarine3') + geom_smooth(method='lm',color='grey80',se=FAl
  theme_cowplot()
p1_n
```



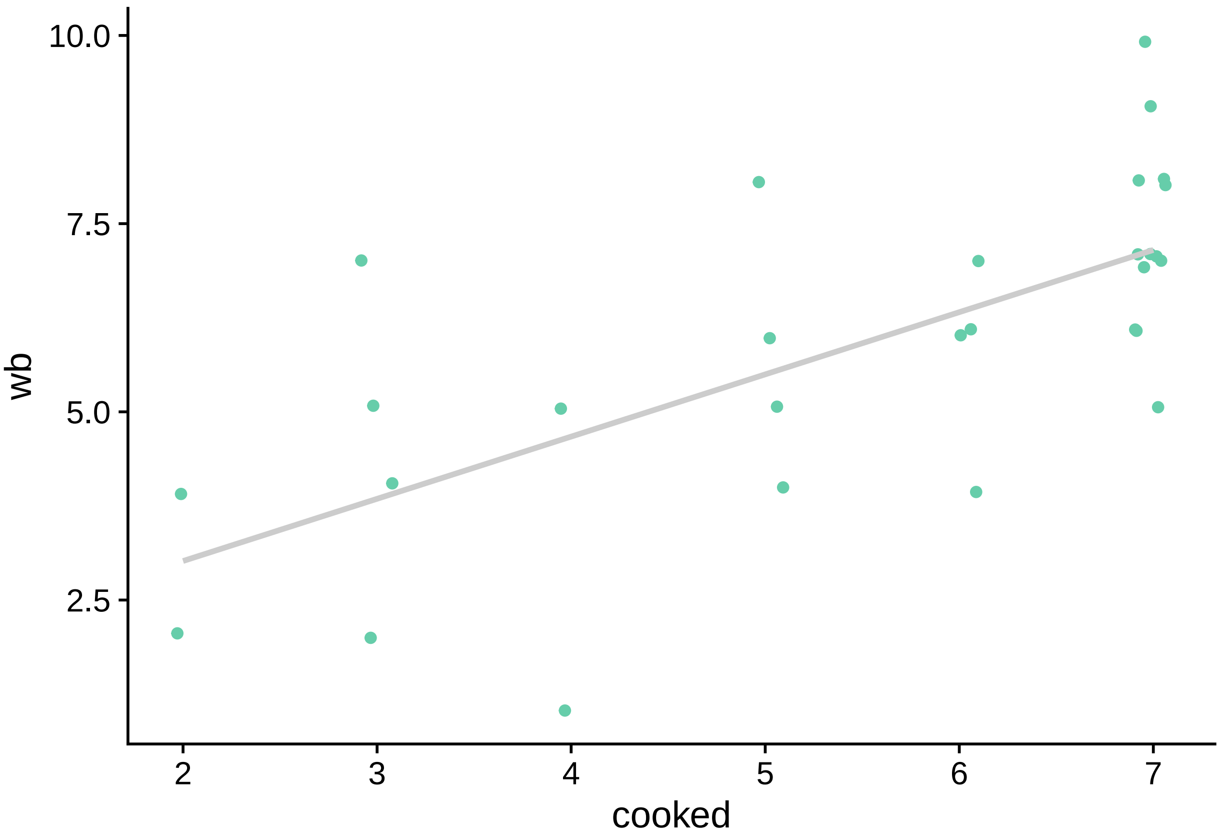
Similarly for Model 2

```
fit2_n <- lm(wb ~ cooked, dw2)
summary(fit2_n)
```

```
##
## Call:
## lm(formula = wb ~ cooked, data = dw2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6711 -1.0182 -0.1505  0.8495  3.1553
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3653     0.9762   1.399    0.173
## cooked        0.8265     0.1699   4.863  4.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.569 on 27 degrees of freedom
## Multiple R-squared:  0.4669, Adjusted R-squared:  0.4472
## F-statistic: 23.65 on 1 and 27 DF,  p-value: 4.399e-05
```

The effect of `cooked` has increased along with its statistical significance.

```
p2_n <- ggplot(dw2, mapping=aes(x=cooked,y=wb)) +
  geom_jitter(height=0.1,width=0.1, color='aquamarine3') + geom_smooth(method='lm',color='grey80',se=FAL
  theme_cowplot()
p2_n
```

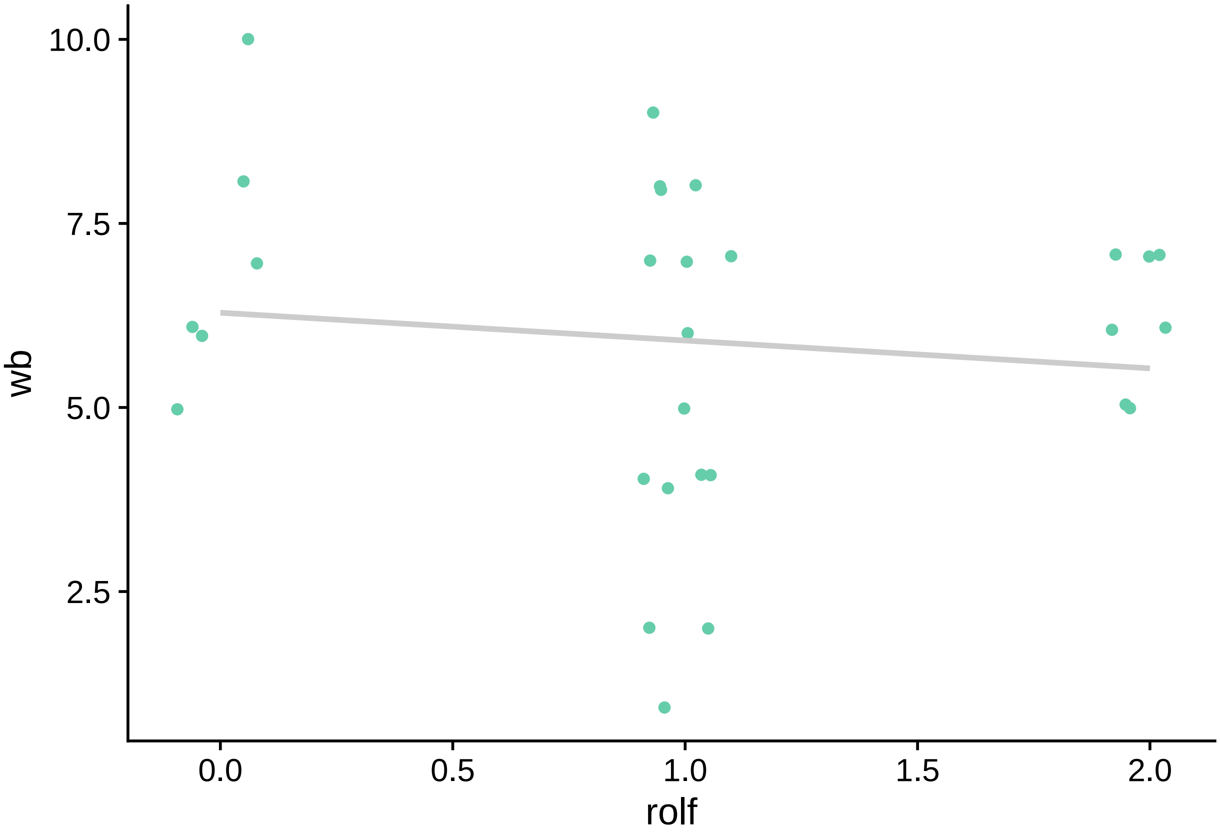


And for Model 3

```
fit3_n <- lm(wb ~ rolf, dw2)
summary(fit3_n)
```

```
##
## Call:
## lm(formula = wb ~ rolf, data = dw2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9096 -1.2872  0.4681  1.4681  3.7128
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.2872     0.7297   8.617 3.13e-09 ***
## rolf         -0.3777     0.5924  -0.638    0.529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.133 on 27 degrees of freedom
## Multiple R-squared:  0.01483,    Adjusted R-squared:  -0.02166
## F-statistic: 0.4065 on 1 and 27 DF,  p-value: 0.5291
```

```
p3_n <- ggplot(dw2, mapping=aes(x=rolf,y=wb)) +
  geom_jitter(height=0.1,width=0.1, color='aquamarine3') + geom_smooth(method='lm',color='grey80',se=FA
  theme_cowplot()
p3_n
```



There is almost no effect of `rolf` on `wb`.

**How strong are the bivariate relationships and which of the independent variables is the best predictor for subjective well-being?**

`cooked` has the strongest bivariate relationship, every additional day in the week where a subject cooks increases wellbeing by 0.82 points. It is also the strongest predictor, explaining 46.69% of the variation in well-being. On the other hand `rolf` has the weakest effect and is the worst predictor.

**The `sigma()` function allows extraction of the value of the residual standard deviation. Do so for all models. What does this tell you?**

For Model 1,

```
sigma(fit1_n)
```

```
## [1] 2.107526
```

For Model 2,

```
sigma(fit2_n)
```

```
## [1] 1.568996
```

For Model 3,

```
sigma(fit3_n)
```

```
## [1] 2.13299
```

Model 2 has the lowest residual standard deviation, which makes it also the most accurate estimator and has the least variability. While Model 3 has the worst accuracy or the most variability.

**Write down the regression equation for the models with instagram and with rolf**

For Model 1:

$$wb = 6.726 - 0.010ig$$

For Model 3:

$$wb = 6.287 - 0.377rolf$$

**What is the null hypothesis and the alternative hypothesis for the instagram model?**

$$H_0 : \beta_1 = 0 \;\; v. \; H_1 : \beta_1 \neq 0$$

With a p-value of $0.31 > 0.1$, we fail to reject the hypothesis that $\beta_1 = 0$

## 1.3 Task 2

Based on the models from task 1, predict the subjective well-being for the following cases:

**For the ig model, predict well-being scores of students with 10, 100, or 200 Instagram followers.**

```
df1 <-  data.frame(ig=c(10,100,200))
predict(fit1_n,newdata=df1)
```

```
##        1        2        3 
## 6.621704 5.681024 4.635825
```

This is probably not very accurate due to the low $R^2$ score and the low coefficient for ig. Further, 200 is outside the normal range for `ig`.

**For the cooked model, predict well-being scores of students with 1, 3, or 6 times cooking.**

```
df2 <- data.frame(cooked=c(1,3,6))
predict(fit2_n,newdata=df2)
```

```
##        1        2        3 
## 2.191748 3.844660 6.324029
```

This might be more accurate with the higher goodness-of-fit.

**For the rolf model, predict well-being for a student who saw Rolf 0, 3, or 7 times.**

```
df3 <- data.frame(rolf=c(0,3,7))
predict(fit3_n,newdata=df3)
```

```
##        1        2        3 
## 6.287234 5.154255 3.643617
```

These predictions are not very trustworthy, as no strong relationship was extracted from the regression, further, no value of `rolf` exceeded 2.

And overall, all the predictions are not very useful with the very limited sample size and could be improved by collecting more observations.

## 1.4 Task 3

Setting new values of `cooked` (0-7) to be predicted for:

```
df4 <- tibble(cooked=seq(0,7,length=100))
```

Getting Probability Interval predictions:

```
PI <- predict(fit2_n,newdata=df4,interval='prediction')
colnames(PI) <- c('fit_p','lwr_p','upr_p')
```

Getting Confidence Interval predictions:

```
CI <- predict(fit2_n,newdata=df4,interval='confidence')
colnames(CI) <- c('fit_c','lwr_c','upr_c')
```

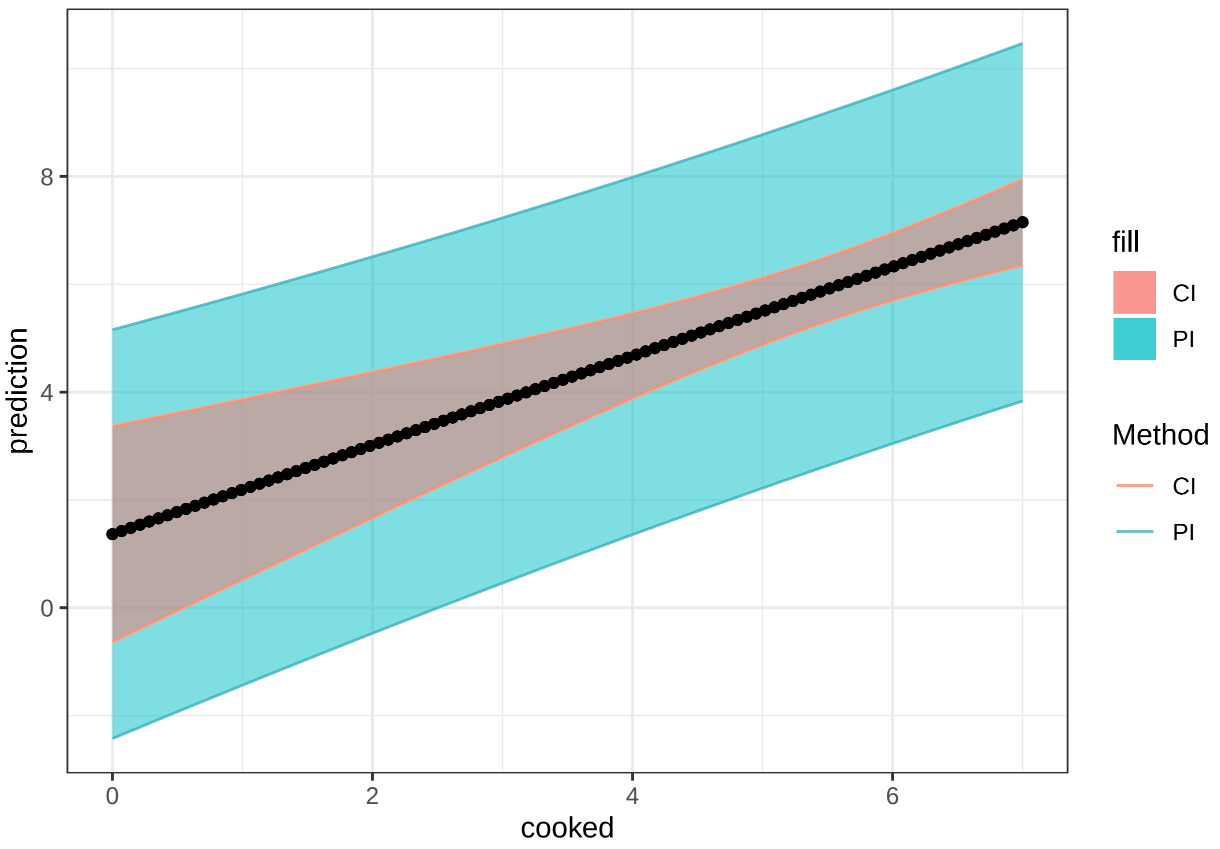
Concatenating both into one dataframe:

```
(results <- as_tibble(cbind(df4,PI,CI)))
```

```
## # A tibble: 100 x 7
##    cooked fit_p lwr_p upr_p fit_c   lwr_c upr_c
##     <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl>
##  1 0       1.37 -2.43  5.16  1.37 -0.638   3.37
##  2 0.0707  1.42 -2.36  5.20  1.42 -0.556   3.40
##  3 0.141   1.48 -2.28  5.25  1.48 -0.474   3.44
##  4 0.212   1.54 -2.21  5.30  1.54 -0.392   3.47
##  5 0.283   1.60 -2.14  5.34  1.60 -0.310   3.51
##  6 0.354   1.66 -2.07  5.39  1.66 -0.228   3.54
##  7 0.424   1.72 -2.00  5.44  1.72 -0.146   3.58
##  8 0.495   1.77 -1.93  5.48  1.77 -0.0647  3.61
##  9 0.566   1.83 -1.86  5.53  1.83  0.0170  3.65
## 10 0.636   1.89 -1.79  5.58  1.89  0.0987  3.68
## # ... with 90 more rows
```

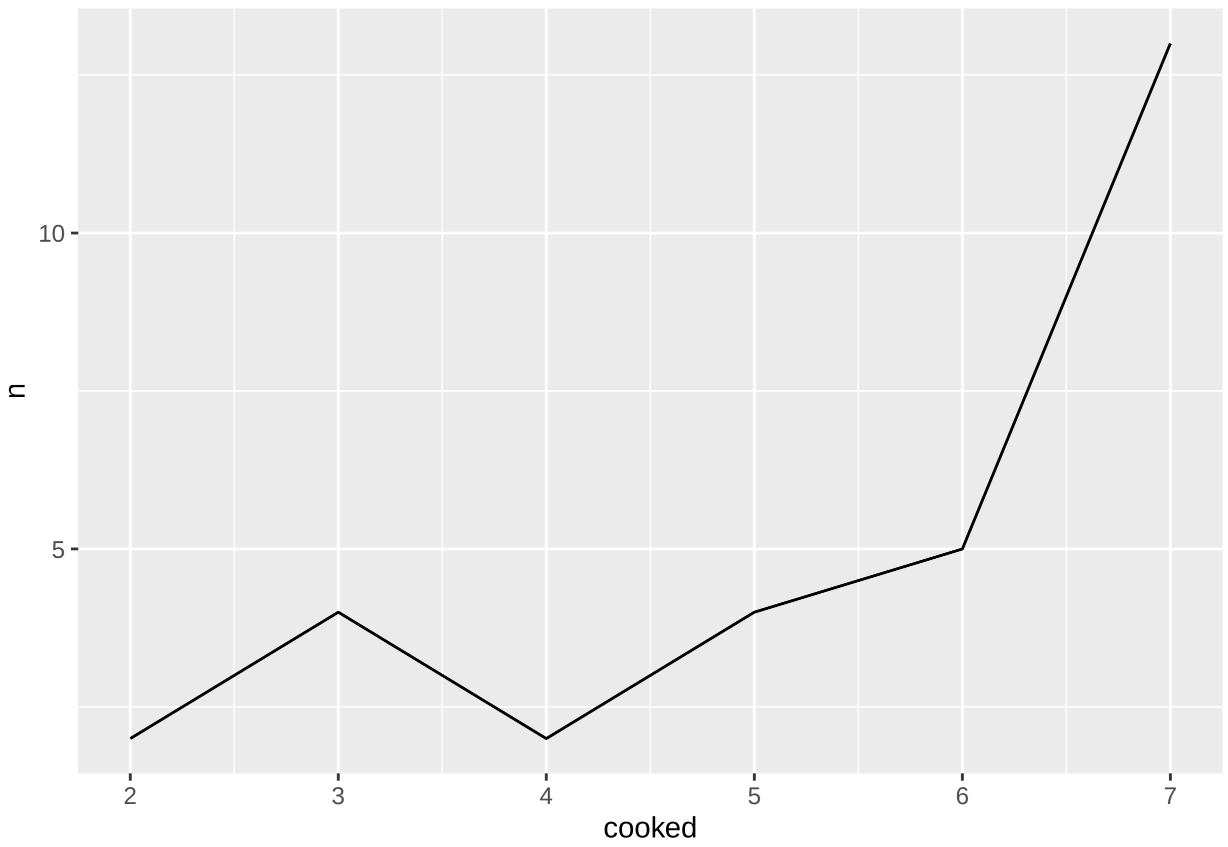
Finally, plotting the results,

```
ggplot(results,mapping=aes(x=cooked)) +
  geom_line(aes(y=lwr_p,color='PI')) +
  geom_line(aes(y=upr_p,color='PI')) +
  geom_ribbon(aes(ymin=lwr_p,ymax=upr_p,fill='PI'),alpha=0.5) +
  geom_line(aes(y=lwr_c,color='CI')) +
  geom_line(aes(y=upr_c,color='CI')) +
  geom_ribbon(aes(ymin=lwr_c,ymax=upr_c,fill='CI'),alpha=0.5) +
  geom_point(aes(y=fit_p)) + geom_line(aes(y=fit_p)) +
  theme_bw() +
  ylab('prediction') +
  scale_color_manual(name='Method',
                     breaks=c('CI','PI'),
                     values=c('CI'='#ffa384','PI'='#74bdcb'))
```



A confidence interval (CI) is an interval containing good estimates of unknown true population parameter. So, the 95% CI means that there is a 95% chance of selecting a sample whose 95% CI contains the true population parameter. Meanwhile, the Prediction Interval (PI) is the interval in which 95% of future observations will occur. Therefore, by definition the PI will always be wider than the CI. Also, note that the CI gets narrower towards the value 7 for `cooked` , this is likely because the number of observations for `cooked` increase with frequency.

```
dw %>% group_by(cooked) %>% summarise(n=n()) %>% ggplot(mapping=aes(x=cooked,y=n)) + geom_line()
```



# 2 PART 2

**Briefly describe some cases (of different types) where it would be inappropriate, or questionable, to use a linear regression model. Provide examples in each case, highlighting the most relevant assumptions.**

- An Least Squares (LS) regression would be inappropriate when the output variable is binary (0 or 1) as fitted values cannot be constrained like that and is possible to be greater than 1 or less than 0, which may not make sense depending on the equation.
- It would also be inappropriate to use LS when estimators aren't linear. For example $\beta_0 + \beta_1 X$ is fine but $\beta_0 + X^{\beta}$ is not able to be estimated using LS
- The explanatory variables should not be significantly linearly correlated, they should be orthogonal. Though, a some multicollinearity is not problematic
- Errors should not be serially correlated, this occurs when there is an omission of some explanatory variables (called false autocorrelation) or if there is a misspecification of the random term (true autocorrelation). OLS is no longer efficient and estimators will have larger variances.
- Heteroskedasticity implies increasing error variances where instead of

  $$var(\epsilon_i) = \sigma^2$$

  the variance has different relations with $X_i$. It implies that OLS is no longer the best estimator, but Weighted Least Squares might be.
- Endogeneity is another major issue, it violates the assumption that regressors are non-stochastic and errors are independent of explanatory variables, and regressors are exogenous. It can occur due to autocorrelated errors, measurement errors, omitted variables, simultaneous equations, or reverse causality.

For example, when studying the effects of drug use on employment probability, it is difficult to isolate the effects of drug use if drug use affects employment probability, but also employment probability can affect drug use. Endogeneity can sometimes be resolved with the use of good Instrumental Variables.