ProjectMarkdown

2022-11-23

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

## Profile

ID = 202249724  
source('XYZprofile.r')  
XYZprofile(ID)

## The profile of XYZ:  
## - Age: 25  
## - Gender: Male  
## - Home address: Nottingham (Colwick)

#### Importing libraries

library(readODS)

## Warning: package 'readODS' was built under R version 4.2.2

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(ggpubr)

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.2

library(ggplot2)  
library(tidyverse)

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ tibble 3.1.8 ✔ purrr 0.3.4  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1  
## ✔ readr 2.1.3 ✔ forcats 0.5.2  
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(CatEncoders)

## Warning: package 'CatEncoders' was built under R version 4.2.2

##   
## Attaching package: 'CatEncoders'  
##   
## The following object is masked from 'package:base':  
##   
## transform

## Cleaning and wrangling the data

datall = read.ods(file = 'dvsa1203.ods')

## Warning in read.ods(file = "dvsa1203.ods"): read.ods will be depreciated in the  
## next version. Use read\_ods instead.

dat = datall[2:16] #The first sheet contains content and metadata.

Looping through our list of dataframe,

cleaned\_data = list()  
for (i in 1:length(dat)) {  
   
 cleaned\_data[[i]] <- dat[[i]][-(1:7),]   
 #Removing metadata; {first 6 rows and the column names(we will assign new ones later)}  
 if (i > 7){ #Removing empty columns in some sheets and fixing their column names  
 cleaned\_data[[i]] <- cleaned\_data[[i]][-c(6,10)]  
 colnames(cleaned\_data[[i]]) = colnames(cleaned\_data[[1]])}  
}

Binding all of the dataframes in the above list in one, and assigning column names

full\_data = bind\_rows(cleaned\_data, .id = 'Year', )  
colnames(full\_data) = c('Year','Centre','Age','Conducted\_Male','Passes\_Male','Pass\_Rate\_Male',  
 'Conducted\_Female','Passes\_Female','Pass\_Rate\_Female',  
 'Conducted\_Total','Passes\_Total','Pass\_Rate\_Total')

Re-encoding missing values of centres

full\_data$Centre[which(full\_data$Centre == "",arr.ind = TRUE)] <- NA  
full\_data <- full\_data %>% fill(Centre, .direction = 'down')

full\_data[which(full\_data == "..",arr.ind = TRUE)] <- NA

Re-encoding the years

for (i in 1:15) {  
 full\_data$Year[which(full\_data$Year == i,arr.ind = TRUE)] <- 2006 + i}

## Exploring the Data

Passrates <- as.numeric(full\_data$Pass\_Rate\_Male)  
mean(Passrates,na.rm = TRUE);var(Passrates,na.rm = TRUE)

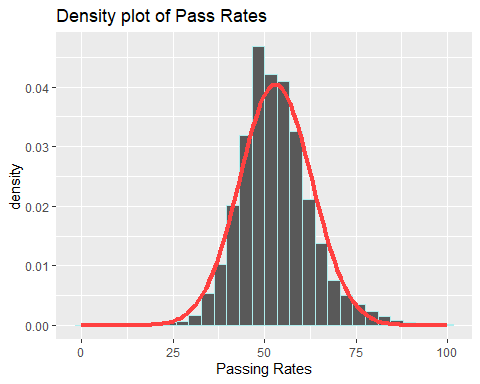
## [1] 53.04242

## [1] 97.28735

ggplot(data.frame(Passrates), aes(x=Passrates)) +  
geom\_histogram(aes(y = after\_stat(density)), col="paleturquoise",na.rm = TRUE) +  
ggtitle("Density plot of Pass Rates") + xlab("Passing Rates") + stat\_function(fun = dnorm, args = list(mean = mean(Passrates,na.rm = TRUE), sd = sd(Passrates,na.rm = TRUE)), lwd=1.5, col="brown1")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggqqplot(Passrates)

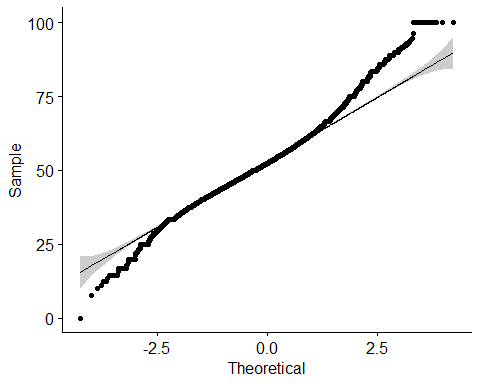
## Warning: Removed 7475 rows containing non-finite values (`stat\_qq()`).

## Warning: Removed 7475 rows containing non-finite values (`stat\_qq\_line()`).

## Warning: The following aesthetics were dropped during statistical transformation: sample  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?

## Warning: Removed 7475 rows containing non-finite values (`stat\_qq\_line()`).

## Warning: The following aesthetics were dropped during statistical transformation: sample  
## ℹ This can happen when ggplot fails to infer the correct grouping structure in  
## the data.  
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical  
## variable into a factor?



Function for getting data of specific city

specity <- function(city){  
 citydata <- data.frame()  
   
 x <- which(full\_data== city,arr.ind = TRUE)  
 p <- full\_data[x[,1],]  
 citydata <- rbind(citydata,p)  
 rownames(citydata) <- NULL  
 citydata <- citydata[citydata$Centre == city,]  
 citydata <- citydata[!(citydata$Age == 'Total' | citydata$Age == ''),]  
 return(citydata)  
}

nearLSE <- specity('Wood Green (London)')

We see that data is missing for years past 9, as city name is different being Wood Green. So, we will combine this data too.

extra <- specity('Wood Green')  
nearLSE <- rbind(nearLSE,extra)

nearhome <- specity('Nottingham (Colwick)')

Function to get data for specific Age

specage <- function(df,age){  
 agedata <- data.frame()  
 x <- which(df$Age== age,arr.ind = TRUE)  
 p <- df[x,]  
 agedata <- rbind(agedata,p)  
 rownames(agedata) <- NULL  
 return(agedata)  
}

LSEaged <- specage(nearLSE,25)

homeaged <- specage(nearhome,25)

library(nortest)  
ad.test(Passrates)

##   
## Anderson-Darling normality test  
##   
## data: Passrates  
## A = 172.91, p-value < 2.2e-16

ks.test(Passrates,'pnorm')

## Warning in ks.test.default(Passrates, "pnorm"): ties should not be present for  
## the Kolmogorov-Smirnov test

##   
## Asymptotic one-sample Kolmogorov-Smirnov test  
##   
## data: Passrates  
## D = 0.99998, p-value < 2.2e-16  
## alternative hypothesis: two-sided

numhome <- data.frame(sapply(nearhome, function(x) as.numeric(as.character(x))))

## Warning in FUN(X[[i]], ...): NAs introduced by coercion

numhome['Centre'] <- rep(1,nrow(numhome))

male <- numhome[,c(1,2,3,5)]  
   
duplicate <- function(year,centre,age,response){  
 expanded <- 1:response  
 repeated <- data.frame('Year' = year,'Centre' = centre, 'Age' = age,'pass\_fail' = expanded)  
}  
expanded\_rows <- Map(f = duplicate,male$Year, male$Centre, male$Age,male$Passes\_Male)  
maled <- do.call(rbind,expanded\_rows)

maled$pass\_fail <- rep(1,length(maled$pass\_fail))

malefail <- numhome[,c(1,2,3)]  
malefail['Fails\_male'] <- numhome$Conducted\_Male - numhome$Passes\_Male

fexpanded\_rows <- Map(f = duplicate,malefail$Year,malefail$Centre,malefail$Age,malefail$Fails\_male)  
malefailed <- do.call(rbind,fexpanded\_rows)  
malefailed$pass\_fail <- rep(0,length(malefailed$pass\_fail))

homelogistic <- rbind(maled,malefailed)

numlse <- data.frame(sapply(nearLSE, function(x) as.numeric(as.character(x))))

## Warning in FUN(X[[i]], ...): NAs introduced by coercion

numlse['Centre'] <- rep(0,nrow(numlse))  
  
lsepasses <- numlse[,c(1,2,3,5)]  
lsexpanded <- Map(f = duplicate,lsepasses$Year,lsepasses$Centre,lsepasses$Age,lsepasses$Passes\_Male)  
lsepassed <- do.call(rbind,lsexpanded)  
lsepassed['pass\_fail'] <- rep(1,nrow(lsepassed))

lsefails <- numlse[,c(1,2,3)]  
lsefails['pass\_fail'] <- numlse$Conducted\_Male - numlse$Passes\_Male  
  
lsefexpanded <- Map(f = duplicate,lsefails$Year,lsefails$Centre,lsefails$Age,lsefails$pass\_fail)   
lsefailed <- do.call(rbind,lsefexpanded)  
lsefailed$pass\_fail <- rep(0,nrow(lsefailed))

lselogistic <- rbind(lsepassed, lsefailed)

logisticdat = rbind(homelogistic,lselogistic)

To adjust for the high value of years affecting the model, we will encode the Years as 1,2,…..15

logisticdat$Year <- as.numeric(factor(logisticdat$Year))

model <- glm(pass\_fail~.,data = logisticdat)

summary(model)

##   
## Call:  
## glm(formula = pass\_fail ~ ., data = logisticdat)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -0.5684 -0.4933 -0.4112 0.4947 0.5971   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.6090856 0.0174409 34.923 < 2e-16 \*\*\*  
## Year -0.0041703 0.0004944 -8.435 < 2e-16 \*\*\*  
## Centre 0.0611881 0.0041574 14.718 < 2e-16 \*\*\*  
## Age -0.0057452 0.0008091 -7.101 1.25e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for gaussian family taken to be 0.2483653)  
##   
## Null deviance: 15013 on 60060 degrees of freedom  
## Residual deviance: 14916 on 60057 degrees of freedom  
## AIC: 86795  
##   
## Number of Fisher Scoring iterations: 2

We can compute a chi-square statistic to test if a model is useful. Chi-Sq statistic : 15013 - 14916 = 97 for 60060 - 60057 = 3 dof

test <- function(age,year,centre){  
 Age <- c(age);Year <- c(year);Centre <- c(centre)  
 test <- data.frame(Year,Age,Centre)  
 return(test)}

c(predict.glm(model,test(25,16,1),se.fit = TRUE),predict.glm(model,test(25,16,0),se.fit = TRUE))

## $fit  
## 1   
## 0.4599189   
##   
## $se.fit  
## [1] 0.006583998  
##   
## $residual.scale  
## [1] 0.4983626  
##   
## $fit  
## 1   
## 0.3987308   
##   
## $se.fit  
## [1] 0.006166066  
##   
## $residual.scale  
## [1] 0.4983626

library(rms)

## Warning: package 'rms' was built under R version 4.2.2

## Loading required package: Hmisc

## Warning: package 'Hmisc' was built under R version 4.2.2

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':  
##   
## src, summarize

## The following objects are masked from 'package:base':  
##   
## format.pval, units

## Loading required package: SparseM

##   
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':  
##   
## backsolve

lrm(pass\_fail~.,data = logisticdat)

## Logistic Regression Model  
##   
## lrm(formula = pass\_fail ~ ., data = logisticdat)  
##   
## Model Likelihood Discrimination Rank Discrim.   
## Ratio Test Indexes Indexes   
## Obs 60061 LR chi2 390.33 R2 0.009 C 0.547   
## 0 30362 d.f. 3 R2(3,60061)0.006 Dxy 0.094   
## 1 29699 Pr(> chi2) <0.0001 R2(3,45040.3)0.009 gamma 0.094   
## max |deriv| 7e-09 Brier 0.248 tau-a 0.047   
##   
## Coef S.E. Wald Z Pr(>|Z|)  
## Intercept 0.4394 0.0702 6.26 <0.0001   
## Year -0.0168 0.0020 -8.43 <0.0001   
## Centre 0.2456 0.0167 14.68 <0.0001   
## Age -0.0231 0.0033 -7.10 <0.0001   
##

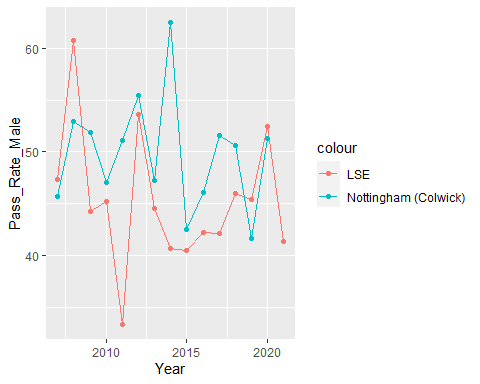
library(ggplot2)  
lseagednum <- data.frame(sapply(LSEaged, function(x) as.numeric(as.character(x))))

## Warning in FUN(X[[i]], ...): NAs introduced by coercion

homeagednum <- data.frame(sapply(homeaged, function(x) as.numeric(as.character(x))))

## Warning in FUN(X[[i]], ...): NAs introduced by coercion

ggplot(data = lseagednum,aes(Year,Pass\_Rate\_Male, col = 'LSE')) + geom\_point() + geom\_line() + geom\_point(data = homeagednum, aes(Year,Pass\_Rate\_Male,col = 'Nottingham (Colwick)')) + geom\_line(data = homeagednum,aes(Year,Pass\_Rate\_Male,col = 'Nottingham (Colwick)'))



## Permutation test

set.seed(1111)  
x <- as.numeric(homeaged$Pass\_Rate\_Male)  
y <- as.numeric(LSEaged$Pass\_Rate\_Male)  
z <- c(x,y)  
stat <- abs(mean(x) - mean(y))   
k <- 0  
for (i in 1:10000) {  
 zperm <- sample(z,29)  
 statperm <- abs(mean(zperm[1:14])-mean(zperm[15:29]))   
 if (statperm > stat) k <- k+1}

pval <- k/10000  
pval

## [1] 0.0479