Data Exploratory Analysis

Simone Zanetti

29/6/2018

```
knitr::opts_chunk$set(message = FALSE, warning = FALSE, results = FALSE, fig.dim=c
(8,4))
```

The following section provides an exploratory analysis of the data available for this project. At first, a generic analysis of the company will be performed to observe the situation and address eventual trends. After that, aggregate data will be created in order to obtain the dependent variable of interest which is the ratio between deliveries and number of worked hours. Each dependent variable is referred to a driver, that is the statistical unit of this analysis. In conclusion, an analysis of the relationship between the y - dependent variable - and each independent variable will be performed in order to analyse the situation.

The following codes need a series of packages to be installed in order to perform the analysis:

```
library(tidyr)
library(dplyr)
library(chron)
library(ggmap)
library(ggplot2)
library(ggthemes)
library(tidyverse)
library(geosphere)

load("Data_exploratory.RData")
```

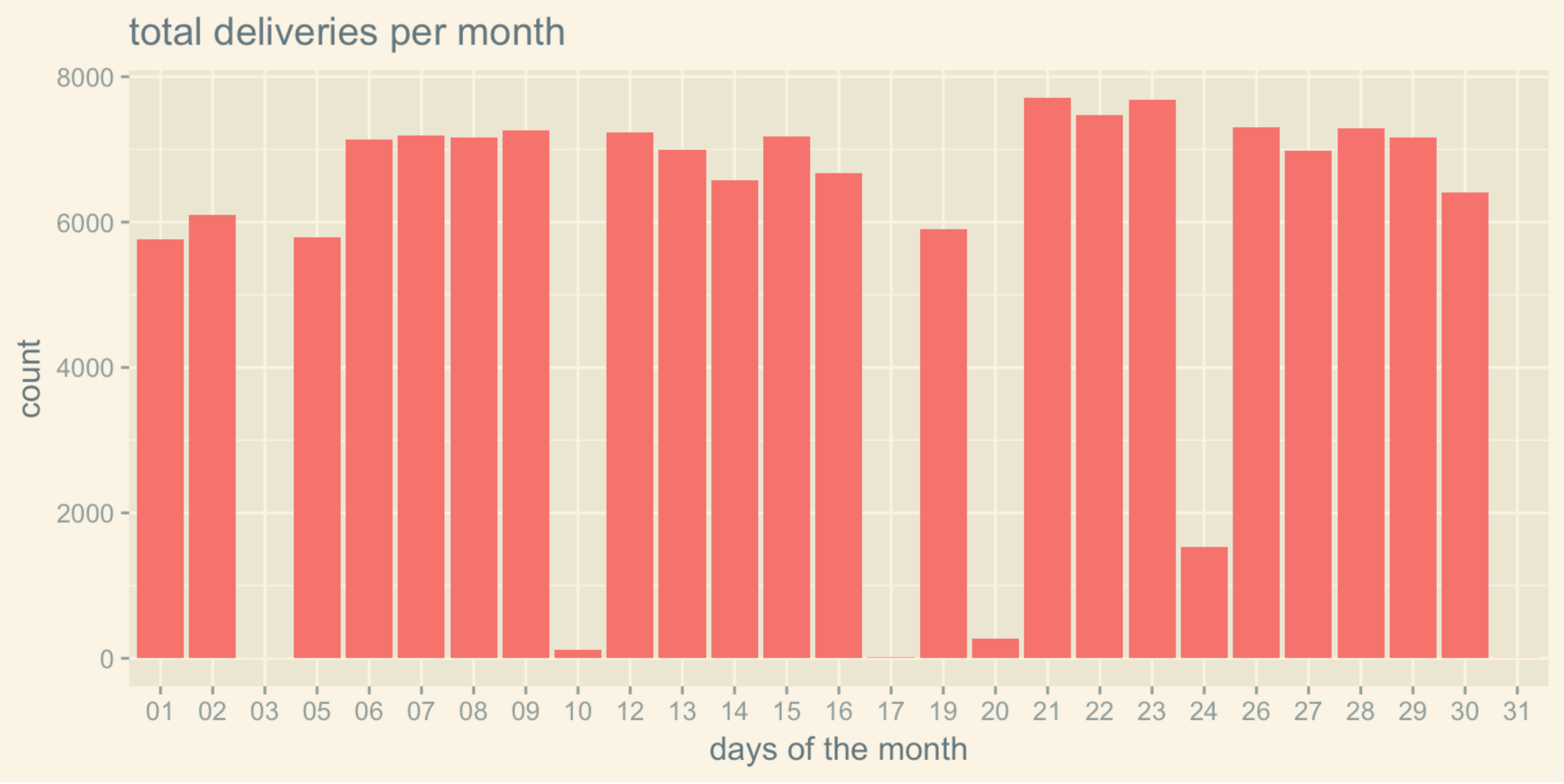
GENERICAL TREND OF THE COMPANY

DAYS OF THE MONTH

The data frame and the plot obtained by the codes below can be summarised as following:

```
data_exploratory$num_pack <- as.numeric(data_exploratory$num_pack)
deliveries_day_driver <- data_exploratory %>% group_by(driver_code,day_deliv) %>%
summarise(tot_deliveries = sum(delivery))
summary(deliveries_day_driver)
```

```
day_plot <- ggplot(data = data_exploratory, aes(x = day_deliv, fill = "indianred2"
)) +
  theme_solarized_2()
day_plot  + geom_bar() +  labs(title = "total deliveries per month", x = "days of
the month") + guides(fill = FALSE)
```



The bar chart shows two aspect of the analysis that need to be fixed: 1) For future analysis based on the day of the week there are one Tuesday and One Friday more than any other weekday. 2) The Tuesday 20 corresponds to a strike that took place in Brescia. As a consequence, deliveries that day have been limited and apparently redistribuited in the following days causing an increase on the number of deliveries the following days.

This issue can cause problems in the moment where an analysis of the day of the week will be performed and as a consequence solutions will be provided in the following section.

```
stronger_day <- deliveries_day_driver %>% group_by(day_deliv) %>% summarise(tot_de
liveries = sum(tot_deliveries))
stronger_day <- stronger_day %>% arrange(desc(tot_deliveries))

head(stronger_day)
tail(stronger_day)
```

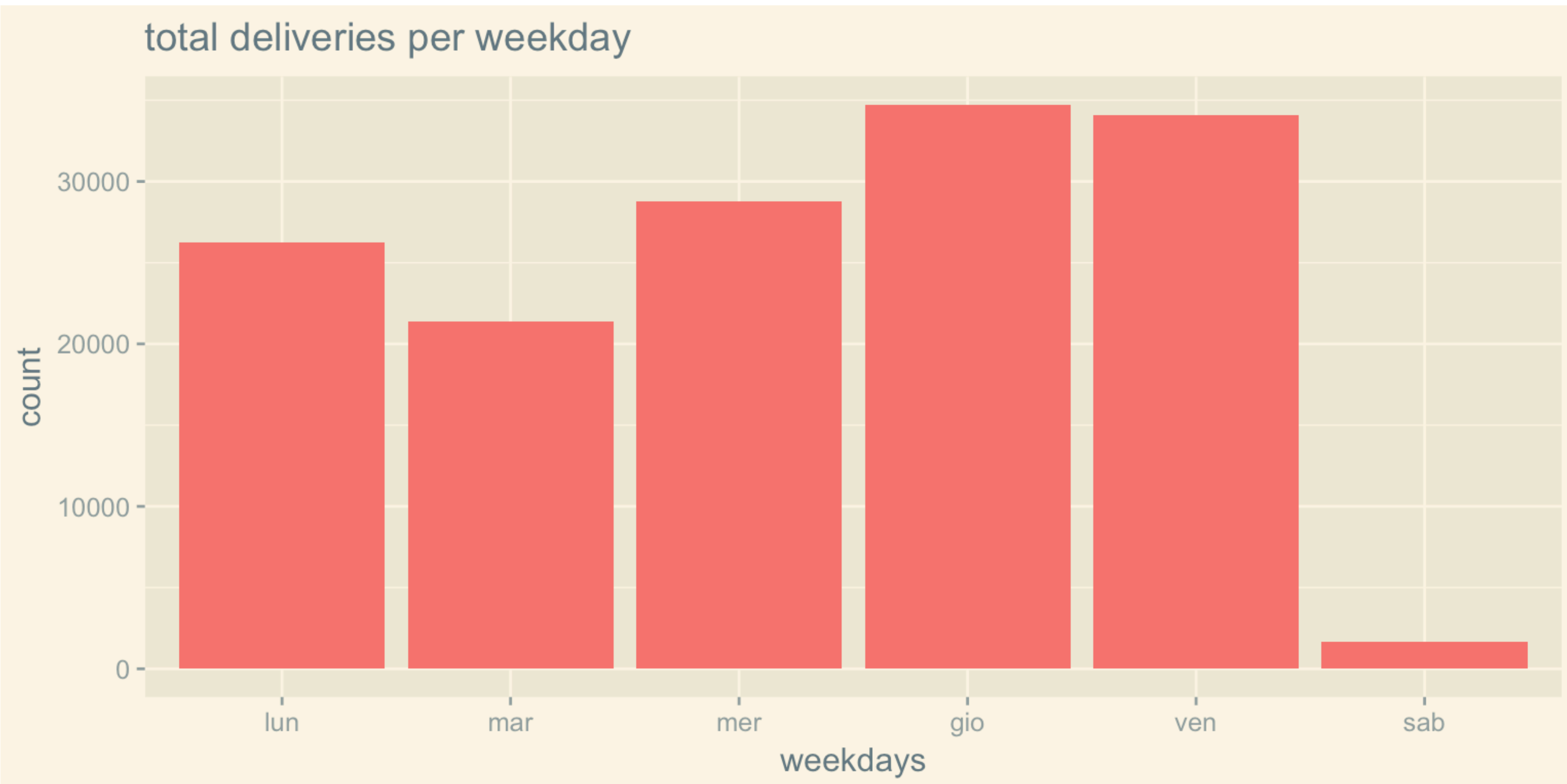
```
summary(stronger_day)
```

The chart shows the 21, 23 and 22 to be the days with more deliveries and this is apparently coherent with the fact that the 20 March a strike paralized the normal activity of the company. The Mean of deliveries per day is 5441 with a Median of 6984 suggesting a distribution skewed to the left.

ANALYSIS OF WEEKDAYS

```
data_exploratory$weekday_deliv <- factor(x = data_exploratory$weekday_deliv, level
s = c("lun","mar","mer","gio","ven","sab"))


stronger_weekday_driver <- data_exploratory %>% group_by(driver_code,weekday_deliv
) %>% summarise(tot_delivery = sum(delivery))

stronger_weekday <- data_exploratory %>% group_by(weekday_deliv) %>% summarise(tot
_delivery = sum(delivery))
```

```
weekday_plot <- ggplot(data = data_exploratory, aes(x = weekday_deliv, fill = "ind
ianred2")) +
  theme_solarized_2()+ labs(title = "total deliveries per weekday", x = "weekdays"
) + guides(fill = FALSE)
weekday_plot + geom_bar()
```

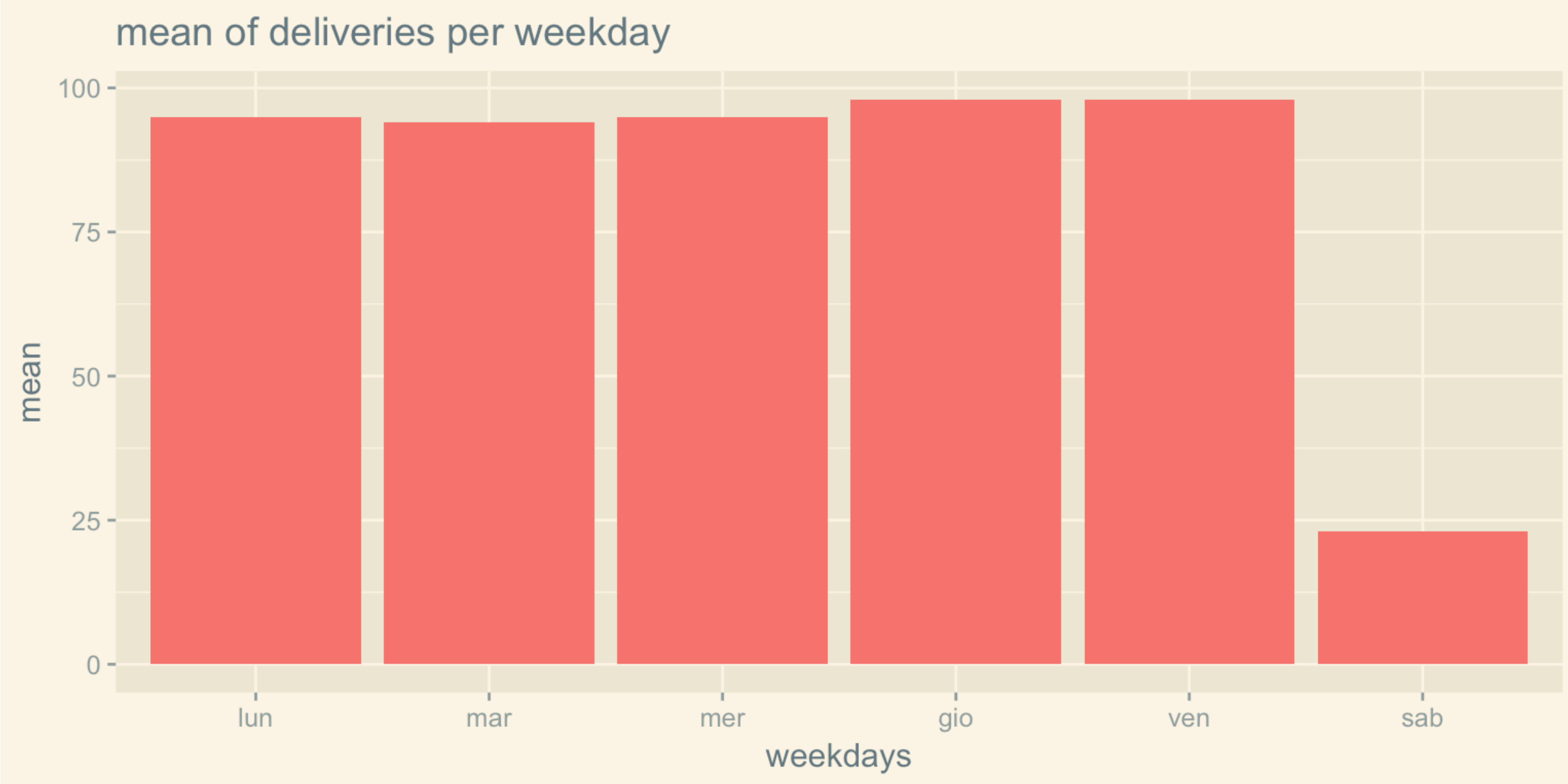


As aforementioned described, these data can be misleading due to the reasons previously described. To overcome this, two possibilites:

1) Use the Mean

```
weekday_mean <- stronger_weekday_driver %>% group_by(weekday_deliv) %>% summarise(
mean = mean(n()))
```

```
weekday_mean_plot <- ggplot(data = weekday_mean, aes(x = weekday_deliv, y = mean,
fill = "indianred2")) +
  theme_solarized_2()
weekday_mean_plot + geom_col()+ labs(title = "mean of deliveries per weekday", x =
"weekdays") + guides(fill = FALSE)
```



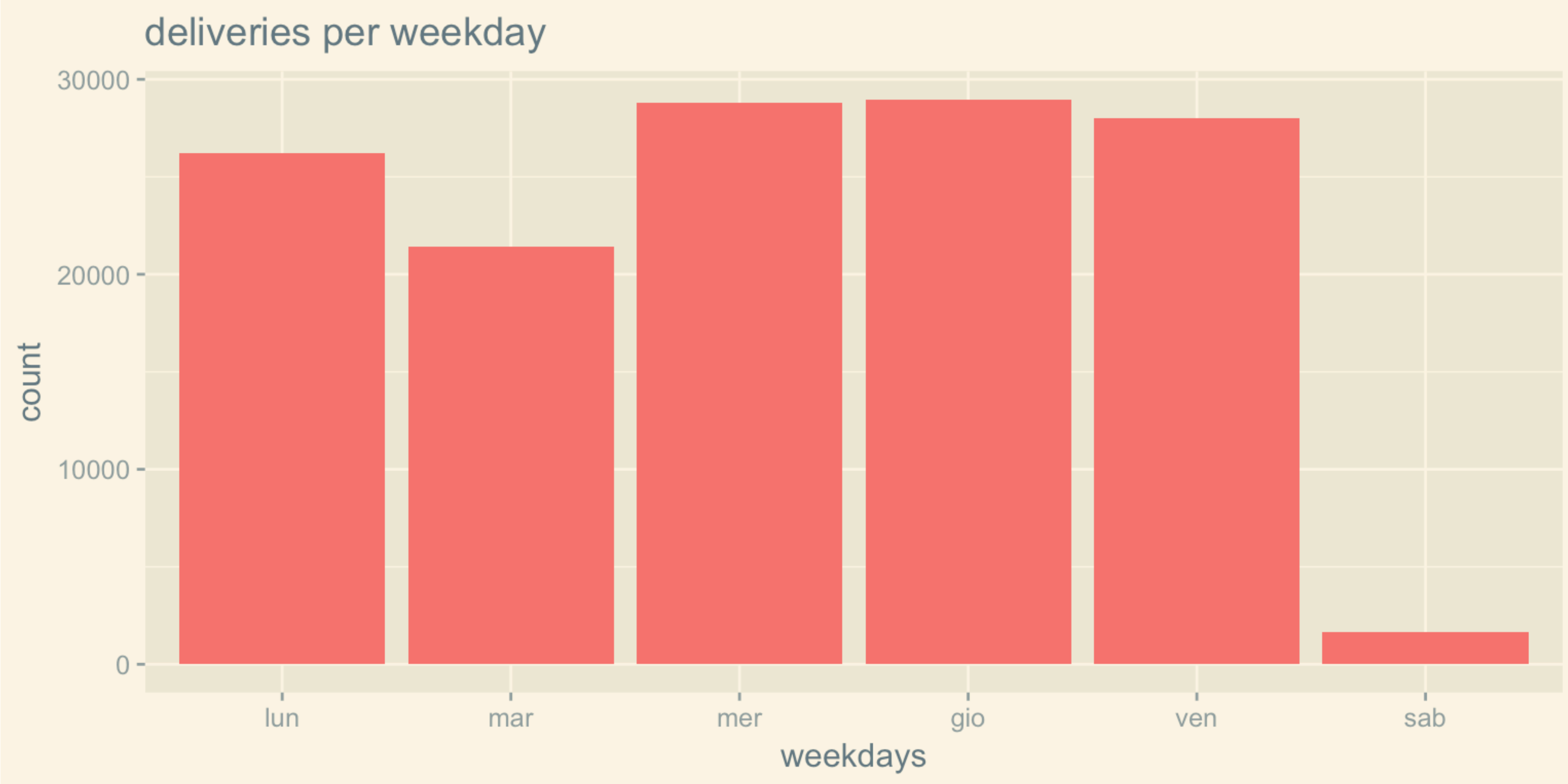
The new graph shows a more leveled situation. The Saturday results the day with less deliveries, as expectable by the fact that the company works only the morning.

2) WORK WITH COMPLETE WEEKS ONLY

I erase the first two days of the month in order to obtain just complete weeks, from Monday to Saturday.

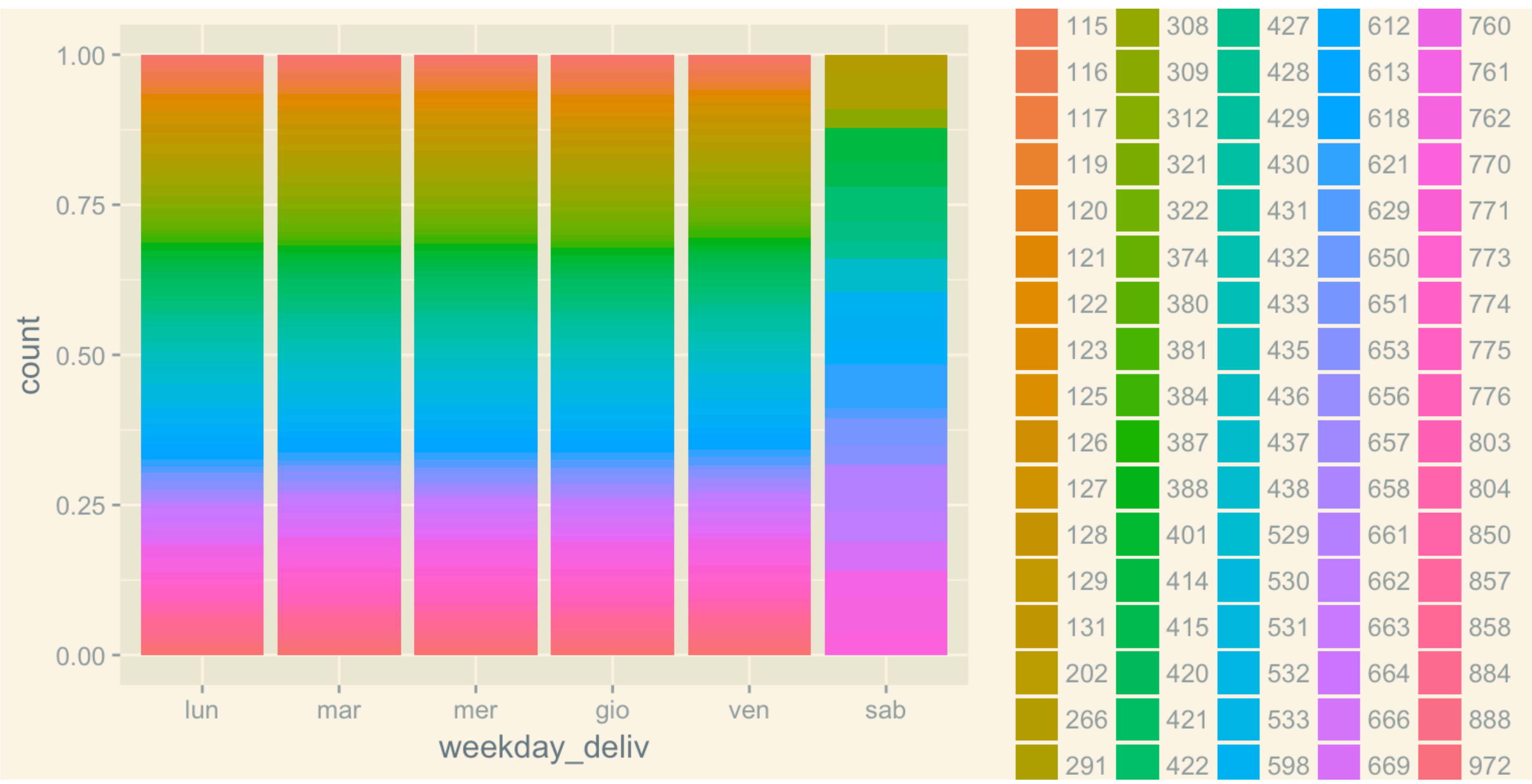
```
data_strongerweekdays <- data_exploratory %>% filter(data_exploratory$day_deliv !=
"01",
                                                     data_exploratory$day_deliv !=
"02",
                                                     data_exploratory$day_deliv !=
"03")
```

```
weekday2_plot <- ggplot(data = data_strongerweekdays, aes(x = weekday_deliv, fill
= "indianred2")) +
  theme_solarized_2()
weekday2_plot + geom_bar()+ labs(title = "deliveries per weekday", x = "weekdays")
+ guides(fill = FALSE)
```

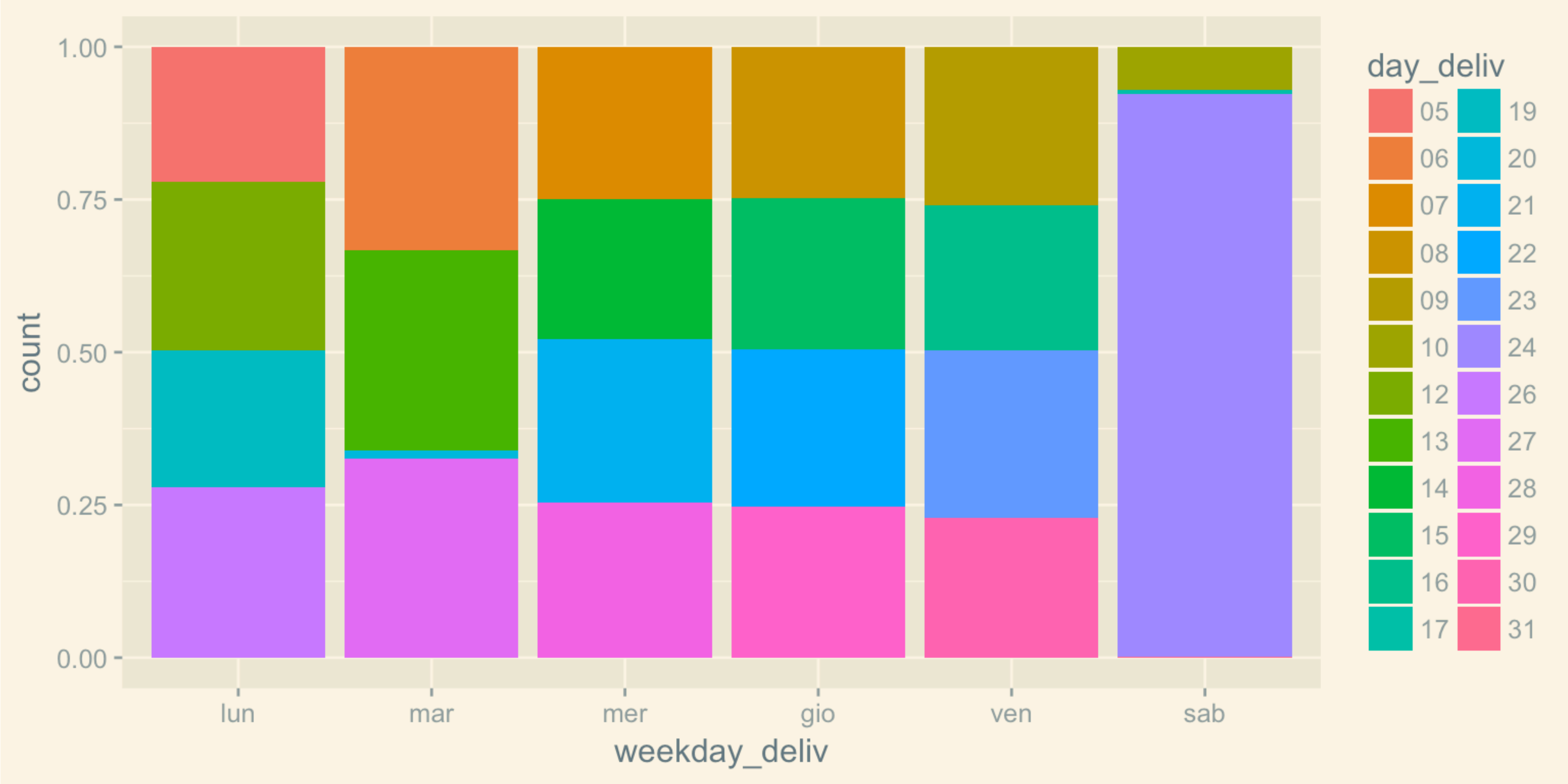


The following codes aimed to deepen the analysis to observe if weekdays can be significantly influenced by other variables such as the driver, the day of delivery and the range of hour.

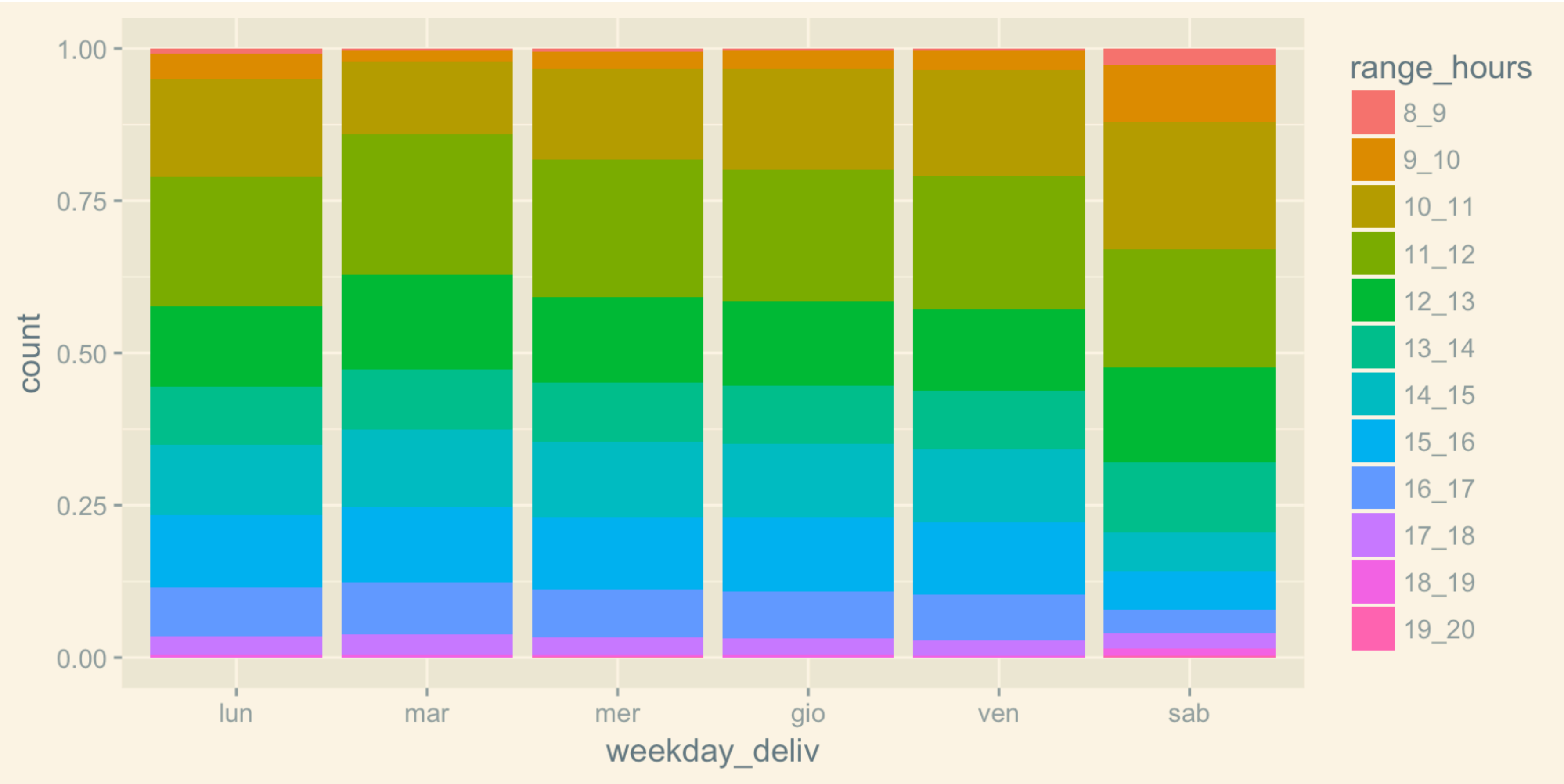
```
weekday2_plot + aes(fill = driver_code) + geom_bar(position = "fill")
```




```
weekday2_plot + aes(fill = day_deliv) + geom_bar(position = "fill")
```




```
weekday2_plot + aes(fill = range_hours) + geom_bar(position = "fill")
```



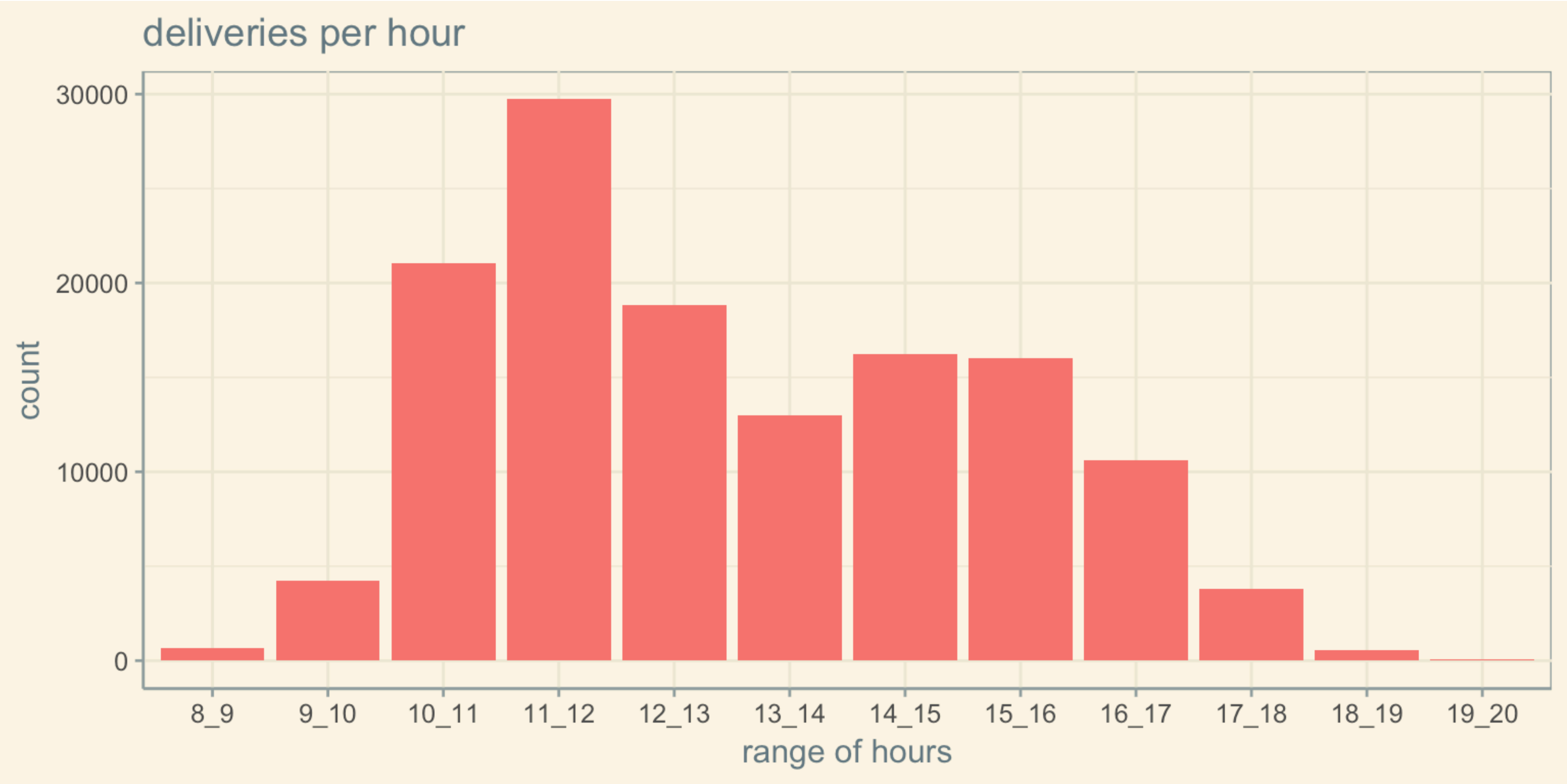
ANALYSIS OF THE TIME OF DELIVERY

Creating of Range Hours from the variable pickedup_time

```
data_exploratory = data_exploratory %>% mutate(range_hours = sub(pattern = "^08.*"
, replacement = "8_9",x = pickup_time),
                                               range_hours = sub(pattern = "^09.*"
, replacement = "9_10",x = range_hours),
                                               range_hours = sub(pattern = "^10.*"
, replacement = "10_11",x = range_hours),
                                               range_hours = sub(pattern = "^11.*"
, replacement = "11_12",x = range_hours),
                                               range_hours = sub(pattern = "^12.*"
, replacement = "12_13",x = range_hours),
                                               range_hours = sub(pattern = "^13.*"
, replacement = "13_14",x = range_hours),
                                               range_hours = sub(pattern = "^14.*"
, replacement = "14_15",x = range_hours),
                                               range_hours = sub(pattern = "^15.*"
, replacement = "15_16",x = range_hours),
                                               range_hours = sub(pattern = "^16.*"
, replacement = "16_17",x = range_hours),
                                               range_hours = sub(pattern = "^17.*"
, replacement = "17_18",x = range_hours),
                                               range_hours = sub(pattern = "^18.*"
, replacement = "18_19",x = range_hours),
                                               range_hours = sub(pattern = "^19.*"
, replacement = "19_20",x = range_hours),
                                               range_hours = sub(pattern = "^20.*"
, replacement = "20_21",x = range_hours))




unique(data_exploratory$range_hours)

data_exploratory = data_exploratory %>% filter(range_hours != "20_21" & range_hour
s != "06:51:00")

data_exploratory$range_hours <- factor(x = data_exploratory$range_hours, levels =
c("8_9","9_10","10_11","11_12","12_13","13_14","14_15","15_16","16_17","17_18","18
_19", "19_20"))
```

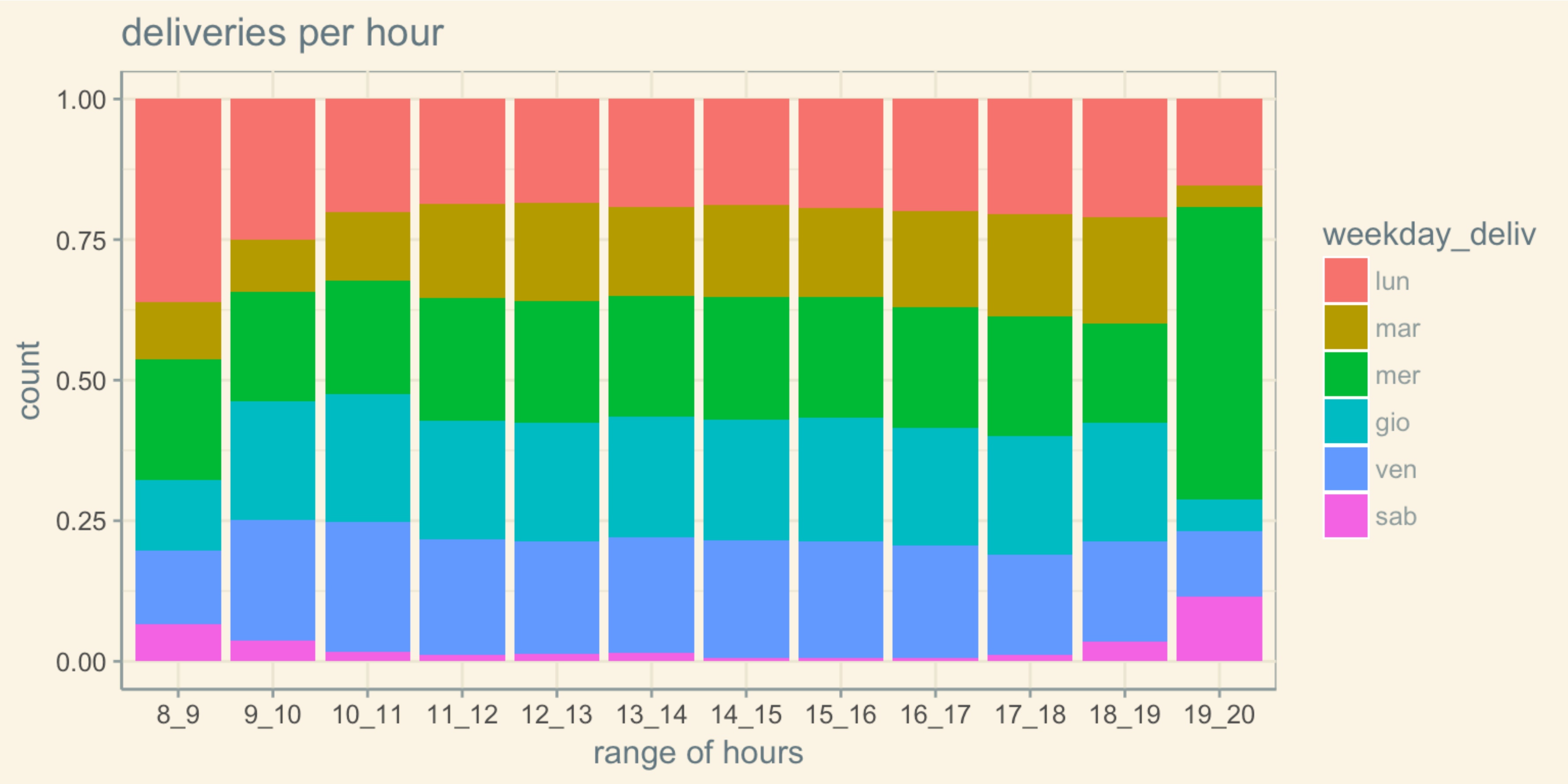
```
stronger_hour <- data_exploratory %>% group_by(driver_code,range_hours) %>% summar
ise(tot_deliveries = sum(delivery))

data_strongerhours <- data_exploratory %>% filter(data_exploratory$day_deliv != "0
1",
                                                  data_exploratory$day_deliv != "0
2",
                                                  data_exploratory$day_deliv != "2
0")
```

```
strongerhour_plot <- ggplot(data = data_strongerhours, aes(x = range_hours, fill=
"indianared2")) + theme_solarized()+
  labs(title = "deliveries per hour", x = "range of hours")

strongerhour_plot + geom_bar() +  guides(fill = FALSE)
```

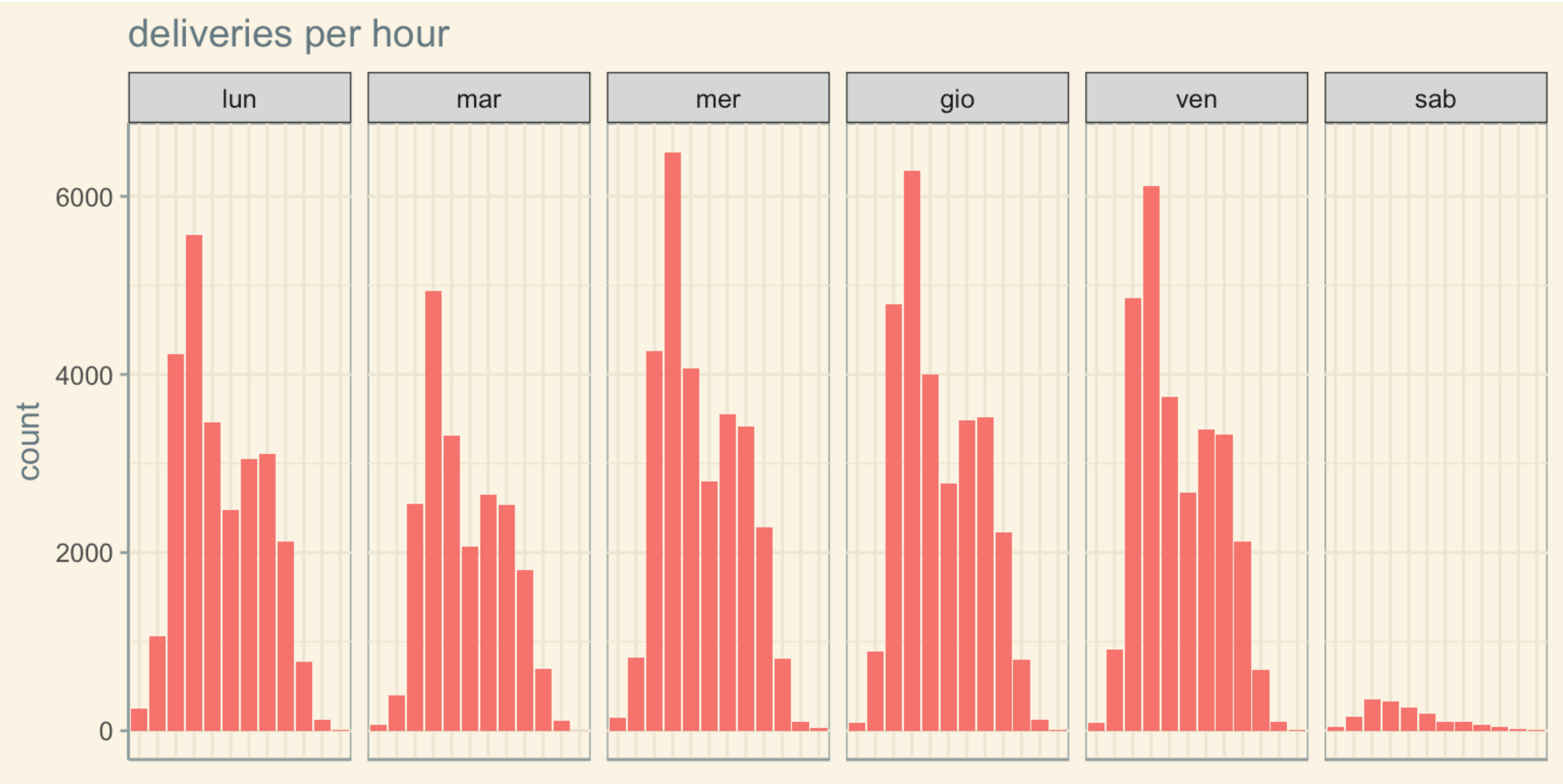


The plot shows a peak of deliveries on the range between 11 am and 12 pm. After that range, the curve regularly decreases until the end of the day with the exepction of the range between 13 and 14, where the number of deliveries decreases before to increase again on the following range. From this point of view, the morning between 10 and 13 the majority of the deliveres are done.

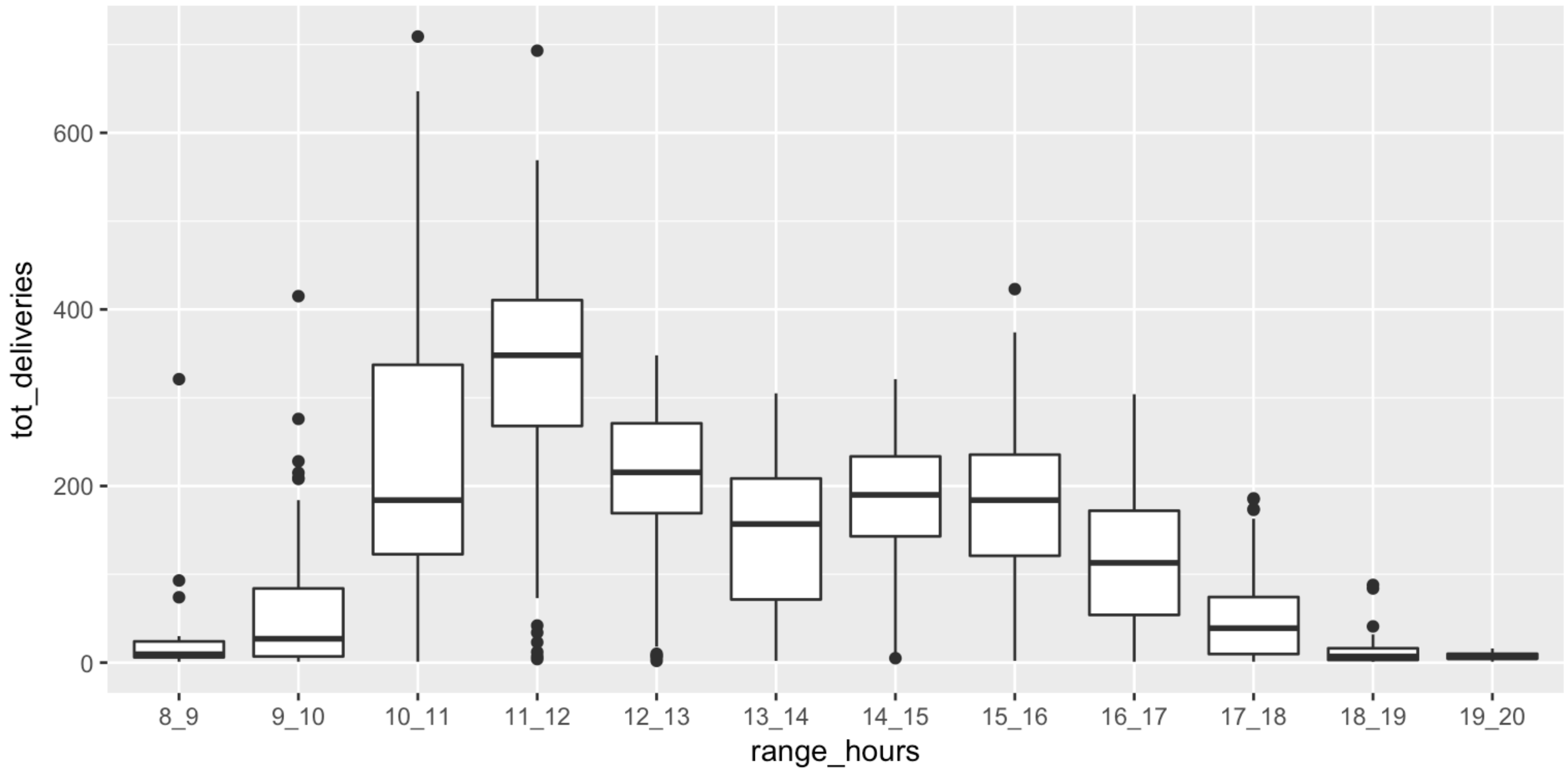
```
strongerhour_plot + aes(fill = weekday_deliv) + geom_bar(position = "fill")
```



```
strongerhour_plot + geom_bar() + facet_grid(.~data_strongerhours$weekday_deliv)+
guides(fill = FALSE)+theme(axis.title.x=element_blank(),
axis.text.x=element_blank(),
axis.ticks.x=element_blank())
```




```
ggplot(data = stronger_hour, aes(x = range_hours, y = tot_deliveries))+ geom_boxpl
ot()
```



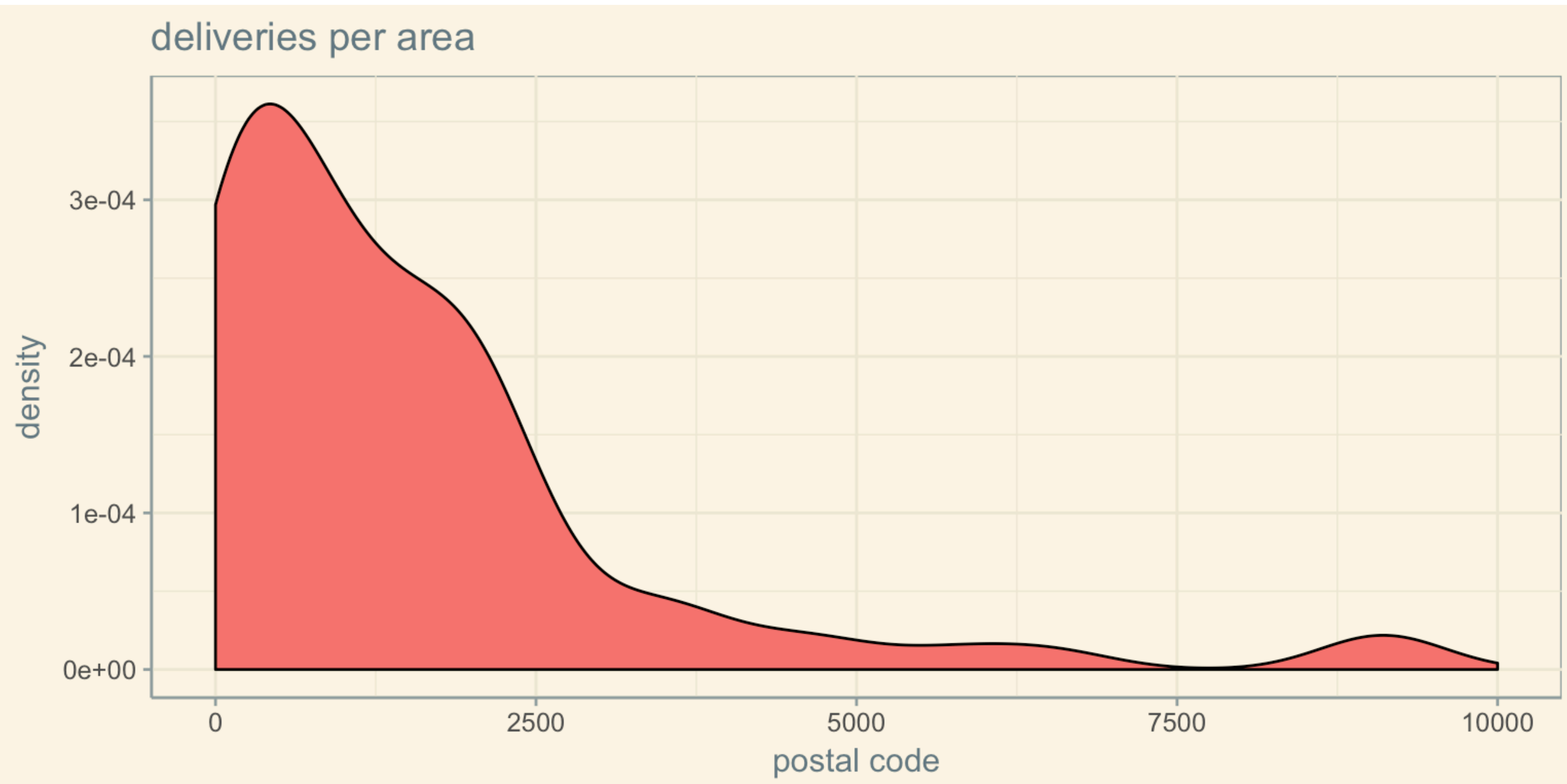
The trend seems to suggest a different attitude of deliveries during the day based on the type of day of the week. In particular, Monday is the day where the majority of deliveries between 8 and 9 am are performed.

ANALYSIS OF THE AREA OF DELIVERY

```
stronger_area <- data_exploratory %>% group_by(postal_code) %>% summarise(cnt = n(
)) %>%
                 arrange(desc(cnt))

summary(stronger_area)
tail(stronger_area)
```

```
head(stronger_area)
```

```
strongerarea_plot <- ggplot(data = stronger_area, aes(x = cnt, fill= "indianared2"
)) + theme_solarized()+
  labs(title = "deliveries per area", x = "postal code")
strongerarea_plot + geom_density()+guides(fill = F)+ scale_x_continuous(limits = c
(0,10000))
```



As shown below, we can observe a right skewed distribution whose Mean of deliveries per Area is 1932.8. However, due to the fact that for the center of Brescia one postal code because was not possible to obtain, it has been identified with the observation “25121/25136”. Thisrepresents an important outlier of this distribution. For this, the number of monthly deliveries is 31846, against the second postal code with maximum number of deliveries which is 9111. This postal code is (an area south of Brescia which includes different important district of the city). Third postal code is 25080 which is an area close to the Lake Garda, where commercial activities are really frequent. In this case, due to the aforementioned outlier the median of 1024.5 is more indicated to describe the tendency of this trend.

SPLIT AREA

The aim of this paragraph is to split postal codes into 3 different areas based on the distance between them and the center of Brescia. As imaginable, one of the factor that can really influence the performance of the drivers is the presence of traffic, and the distance of the points of delivery between each others. From this point of view, it will be expected to have more traffic and poins of delivery closer to each other in the center of Brescia and the opposite further from the city center.

At first, I will have to obtain the coordinates of each postal code.

```
experiment <- data_exploratory %>% group_by(postal_code) %>% summarise()
experiment$city <- "Brescia"
experiment <- unite(data = experiment, c(experiment$postal_code, experiment$city),
sep = "-")
experiment <- experiment %>% rename("postal_code" = `c(experiment$postal_code, exp
eriment$city)`)
latlong <- geocode(experiment$postal_code)
experiment <- cbind(experiment, latlong)

experimentNA1 <- filter(experiment, is.na(experiment$lon))
experimentNA1 <- experimentNA1 %>% select(postal_code)
latlong1 <- geocode(experimentNA1$postal_code)
experimentNA1 <- cbind(experimentNA1, latlong1)

experimentNA2 <- filter(experimentNA1, is.na(experimentNA1$lon))
experimentNA2 <- experimentNA2 %>% select(postal_code)
latlong2 <- geocode(experimentNA2$postal_code)
experimentNA2 <- cbind(experimentNA2, latlong2)

experimentNA3 <- filter(experimentNA2, is.na(experimentNA2$lon))
experimentNA3 <- experimentNA3 %>% select(postal_code)
latlong3 <- geocode(experimentNA3$postal_code)
experimentNA3 <- cbind(experimentNA3, latlong3)

# Very often geocode function gives NA value, so that you have to evaluate the fun
ction again in order to work
# For this reason I have created several dataframe in order to have the possibilit
y to evaluate the rows where geocode previously returned NA.

experimentNA4 <- filter(experimentNA3, is.na(experimentNA3$lon))
experimentNA4 <- experimentNA4%>% select(postal_code)
latlong4 <- geocode(experimentNA4$postal_code)
experimentNA4 <- cbind(experimentNA4, latlong4)

experimentNA5 <- filter(experimentNA4, is.na(experimentNA4$lon))
experimentNA5 <- experimentNA5%>% select(postal_code)
latlong5 <- geocode(experimentNA5$postal_code)
experimentNA5 <- cbind(experimentNA5, latlong5)

experimentNA6 <- filter(experimentNA5, is.na(experimentNA5$lon))
experimentNA6 <- experimentNA6%>% select(postal_code)
latlong6 <- geocode(experimentNA6$postal_code)
experimentNA6 <- cbind(experimentNA6, latlong6)

experiment <- experiment %>% filter(!is.na(experiment$lat))
experimentNA1 <- experimentNA1 %>% filter(!is.na(experimentNA1$lat))
experimentNA2 <- experimentNA2 %>% filter(!is.na(experimentNA2$lat))
experimentNA3 <- experimentNA3 %>% filter(!is.na(experimentNA3$lat))
experimentNA4 <- experimentNA4 %>% filter(!is.na(experimentNA4$lat))
```

```
postalcode_latlong <- rbind(experiment,experimentNA1,experimentNA2)

# postalcode_latlong1 <- postalcode_latlong %>%
#                       mutate(postal_code = gsub(pattern = "Brescia", "", postalcod
e_latlong))

postalcode_latlong <- separate(postalcode_latlong, col = postal_code, into= c("pos
tal_code","city"),sep = "-")
postalcode_latlong$city <- NULL
data_exploratory <- left_join(x = data_exploratory, y = postalcode_latlong, by = "
postal_code" )
data_exploratory1 <- NULL
```

Secondly, I will obtain the distances between points and the center of Brescia.

```
postalcode_latlong <- load("postalcode_latlong.RData")
```

```
postalCodes <- postalcode_latlong

#Find co-ordinates of center of Brescia
Brescia <- geocode("Corso Zanardelli, Brescia")

#In case you go over the limit, you can hard code it
Brescia$lon <- 10.220992
Brescia$lat <- 45.5366031

#Calculate Distance between Center and other points
#First create an empty vector to store the results

Distance <- vector("numeric",length = nrow(postalCodes))

#Now claculate the distance using a for loop
for(i in 1:nrow(postalCodes)){
  Distance[i] <- distm(c(Brescia$lon,Brescia$lat),
                       c(postalCodes$lon[i], postalCodes$lat[i]),
                       fun = distHaversine)
}

#Combine it with postalCodes
postalCodes$Distance <- Distance
```

Finally, I will obtain the three different areas of the city storing them in the variable “area”

```
# Calculating the distance using a for loop

area <- vector("character",length = nrow(postalCodes))

for(i in 1:nrow(postalCodes)){
  if(postalCodes$Distance[i] <= 15000)
  {area[i] = "Zone1"}
  else if(postalCodes$Distance[i] > 15000 & postalCodes$Distance[i] <= 30000 )
  {area[i] = "Zone2"}
  else
  {area[i] = "Zone3"}
}

postalCodes$area <- area

table(data_exploratory$area)

data_exploratory <- left_join(x = data_exploratory, y = postalCodes, by = "postal_
code" )

data_exploratory <- data_exploratory %>% select(-lat.y,-lon.y, Distance.y) %>% ren
ame("lon" = lon.x, "lat" = lat.x, "distance" = Distance.x)



data_exploratory <- data_exploratory %>% mutate(zone1 = ifelse(data_exploratory$ar
ea == "Zone1",1,0),
                                            zone2 = ifelse(data_exploratory$area =
= "Zone2",1,0),
                                            zone3 = ifelse(data_exploratory$area =
= "Zone3",1,0))
zone_split <- data_exploratory %>% group_by(driver_code) %>% summarise(del_zone1 =
sum(zone1),del_zone2 = sum(zone2),del_zone3 = sum(zone3))
```

ANALYSIS OF INDEPENDENT VARIABLE Y

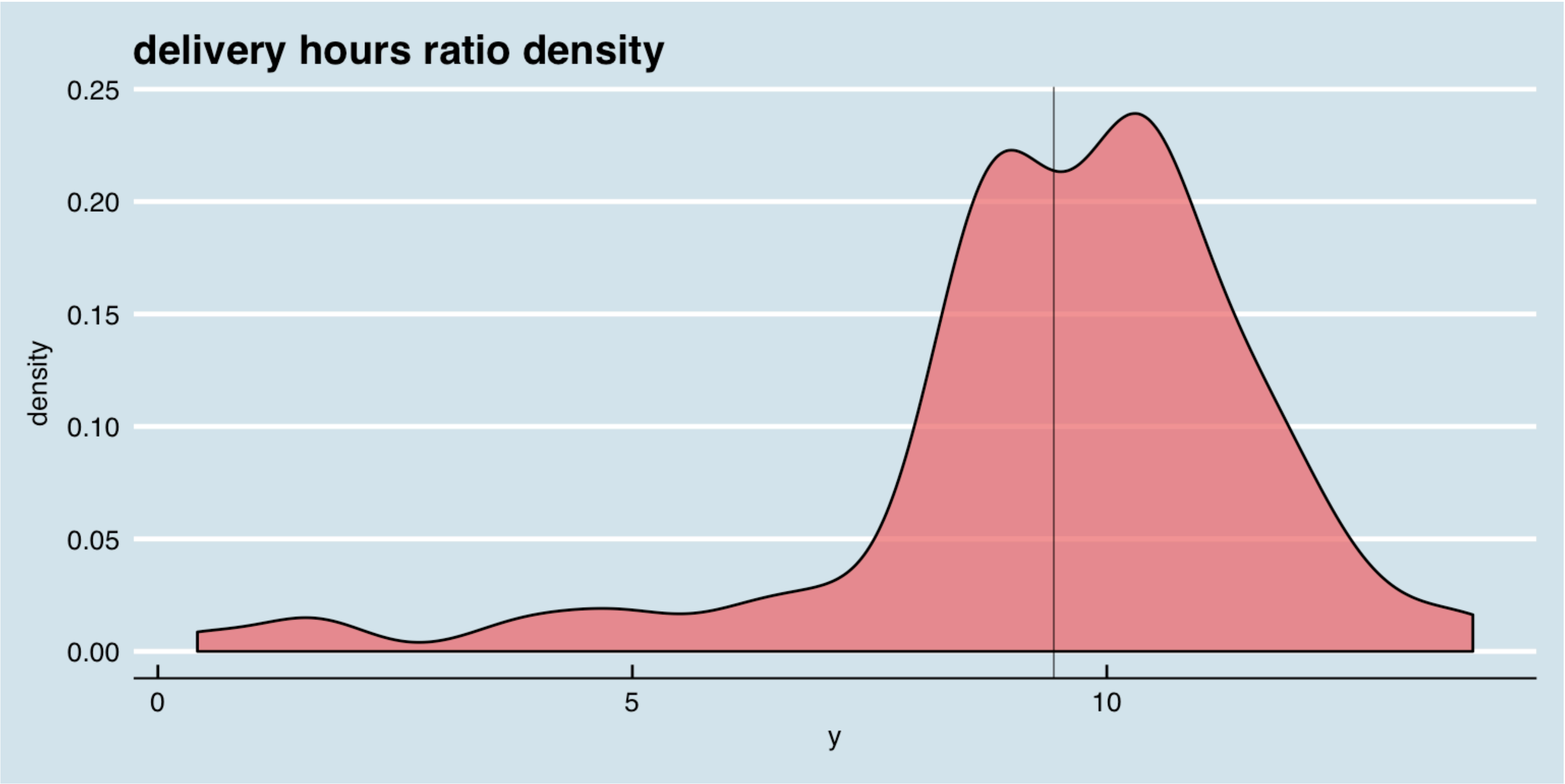
In the following section I will aggregate data in order to obtain the independent variable of this analysis, which is the relationship between deliveries and hours worked for each driver, that represent my statistical unit. By the end of this paragraph a data frame named aggregate_data_last will be obtained in order to conduct a machine learning analysis later on.

OBTAIN THE HOURS WORKED FOR EACH DRIVER AND RATIO DELIVERIES PER DAY WORKED

```
data_exploratory <- na.omit(data_exploratory)
data_exploratory <- data_exploratory %>% filter(driver_code != "208" & driver_code
!= "234"& driver_code != "260" & driver_code != "336" & driver_code != "404" & dri
ver_code != "534" & driver_code != "535" & driver_code != "623"  & driver_code !=
"132")
aggregate_data <- data_exploratory %>% group_by(driver_code, day_deliv, pickup_tim
e, delivery) %>% summarise()

# aggregate_data <- aggregate_data %>% filter(driver_code != "208" & driver_code !
= "234"& driver_code != "260" & driver_code != "336" & driver_code != "404" & driv
er_code != "534" & driver_code != "535" & driver_code != "623"  & driver_code != "
132")

aggregate_data$pickup_time <- as.character(aggregate_data$pickup_time)
aggregate_data$pickup_time = as.POSIXlt(aggregate_data$pickup_time, format = "%H:%
M:%S")

class(aggregate_data$pickup_time)


time_delivery <- rep(0,nrow(aggregate_data))

# Turning time into the difference of minutes between every point of delivery for
every driver

for (i in 1:(nrow(aggregate_data)-1)) {
  if(aggregate_data$driver_code[i] == aggregate_data$driver_code[i+1])
  {time_delivery[i] = difftime(aggregate_data$pickup_time[i], aggregate_data$picku
p_time[i+1], units = "mins" ) }
}

time_delivery=ifelse(time_delivery>0,0,time_delivery)
aggregate_data=aggregate_data[,-3]
aggregate_data=cbind(aggregate_data,time_delivery)
aggregate_data$time_delivery <- abs(aggregate_data$time_delivery)


# sum the worked minutes per driver(minutes worked) and create a new variable base
d on hours (hours worked per driver)

aggregate_data2 <- aggregate_data %>% group_by(driver_code) %>% summarise(tot_deli
veries = sum(delivery), tot_minutes = sum(time_delivery))
hours_delivery <- aggregate_data2$tot_minutes/60
aggregate_data2 <- cbind(aggregate_data2, hours_delivery)

y = aggregate_data2$tot_deliveries / aggregate_data2$hours_delivery
aggregate_data2 <- cbind(aggregate_data2, y)

# aggregate_data2 <- aggregate_data2 %>% filter(driver_code != "208" & driver_code
!= "234"& driver_code != "260" & driver_code != "336" & driver_code != "404" & dri
ver_code != "534" & driver_code != "535" & driver_code != "623"  & driver_code !=
"132")

# Now that I have the total of deliveries per driver and the Independent variable
```

```
I can plot them to analyse the trend.
```

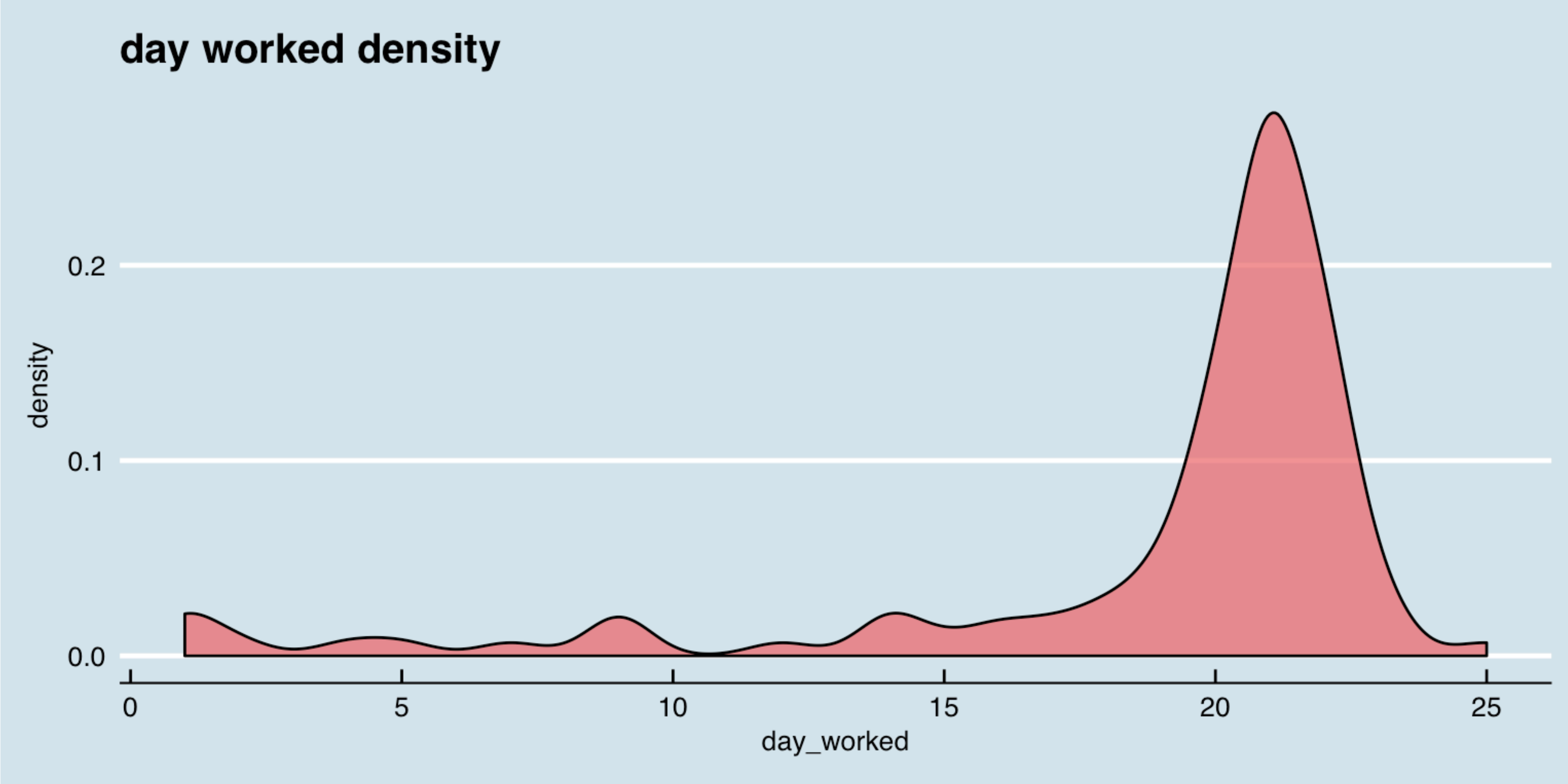
```
summary(aggregate_data2$y)
y_plot <- ggplot(data = aggregate_data2, aes(x = y))
y_plot + geom_density(fill = "indianred2", alpha = 0.7) + theme_economist() + labs
(title = "delivery hours ratio density")+ geom_vline(xintercept = 9.4412, size = 0
.2)
```



The dependent variable y is represented by a left tailed distribution, with a mean of 9.4412 deliveries per hours. 68% of the drivers owns a delivery hours ratio between …

```
aggregate_data$day_deliv <- as.numeric(aggregate_data$day_deliv)
aggregate_data3 <- data_exploratory %>% group_by(driver_code) %>% summarise(day_wo
rked = n_distinct(day_deliv))
# aggregate_data3 <- aggregate_data3 %>% filter(driver_code != "208" & driver_code
!= "234"& driver_code != "260" & driver_code != "336" & driver_code != "404" & dri
ver_code != "534" & driver_code != "535" & driver_code != "623"  & driver_code !=
"132")
aggregate_data2 <- cbind(aggregate_data2, aggregate_data3$day_worked)
```

```
dayworked_plot <- ggplot(data = aggregate_data3, aes(x = day_worked))
dayworked_plot + geom_density(fill = "indianred2", alpha = 0.7) + theme_economist(
) + labs(title = "day worked density")
```



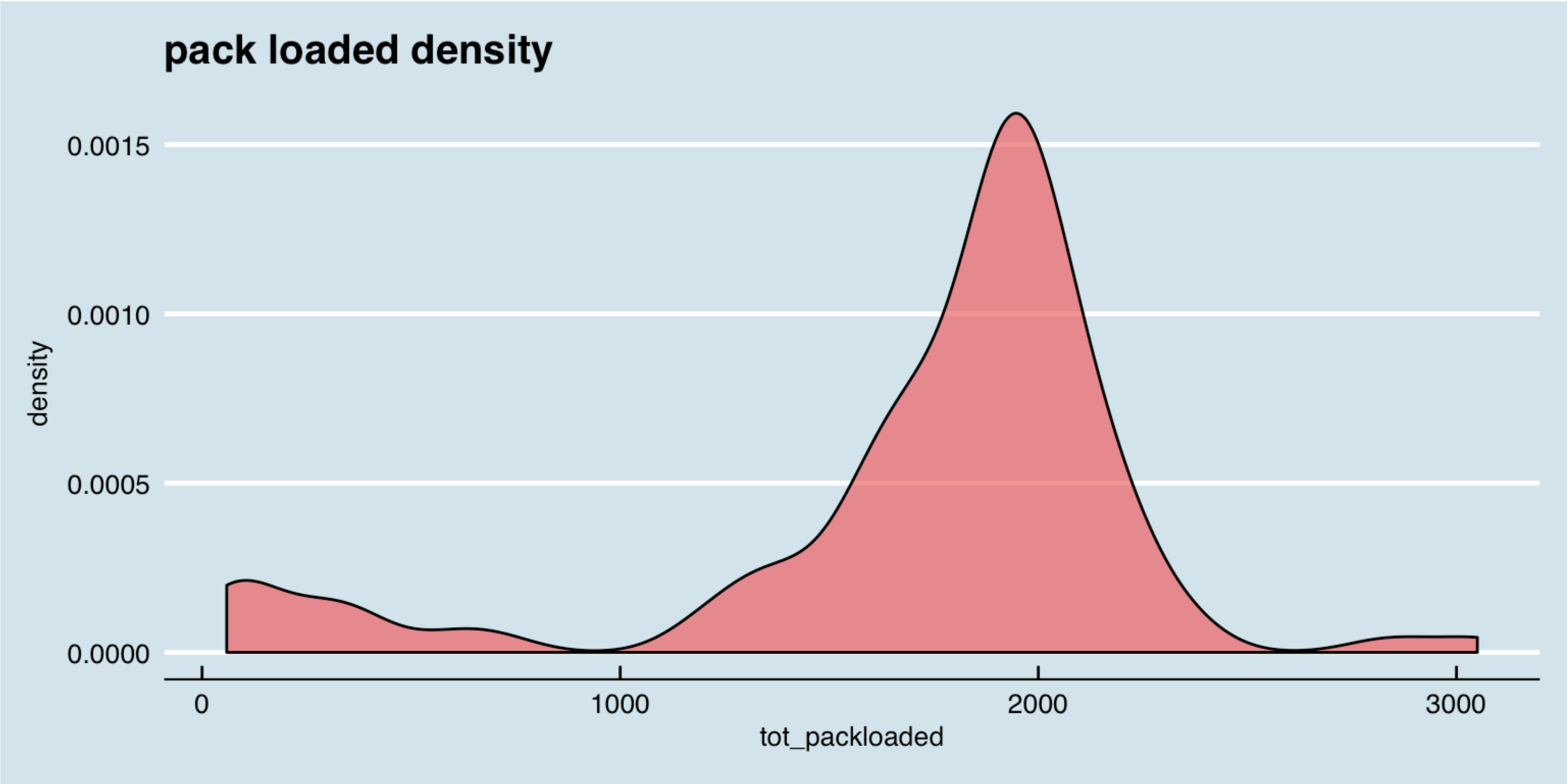
The curve …

TOTAL PACK LOADED PER DRIVER

```
load("aggregate_data4.Rdata") # Load a dataset previously cleaned during the wrang
ling phase. Dataset provided by the client later on

# clean the data for the needs of this analysis
aggregate_data4 <- aggregate_data4 %>% filter(driver_code != "208" & driver_code !
= "234"& driver_code != "260" & driver_code != "336" & driver_code != "404" & driv
er_code != "534" & driver_code != "535" & driver_code != "623"  & driver_code != "
132",
                                              driver_code != "1000", driver_code !
= "101", driver_code != "402")
aggregate_data4 <- aggregate_data4 %>% mutate(driver_code = gsub(pattern = "8421",
replacement = "421",x = driver_code),
                                              driver_code = gsub(pattern = "8618",
replacement = "618", x = driver_code),
                                              driver_code = gsub(pattern = "8678",
replacement = "678", x = driver_code),
                                              driver_code = gsub(pattern = "8679",
replacement = "679", x = driver_code),
                                              driver_code = gsub(pattern = "8531",
replacement = "531", x = driver_code))
```

```
aggregate_data5 <- aggregate_data4 %>% group_by(driver_code ) %>% summarise(tot_pa
ckloaded = sum(pack_loaded))
setdiff(aggregate_data5$driver_code, aggregate_data2$driver_code)
aggregate_data5 <- aggregate_data5 %>% filter(tot_packloaded > 0, driver_code != "
851", driver_code != "805")
```

```
packloaded_plot <- ggplot(data = aggregate_data5, aes(x = tot_packloaded))
packloaded_plot +  geom_density(fill = "indianred2", alpha = 0.7) + theme_economis
t() + labs(title = "pack loaded density")
```

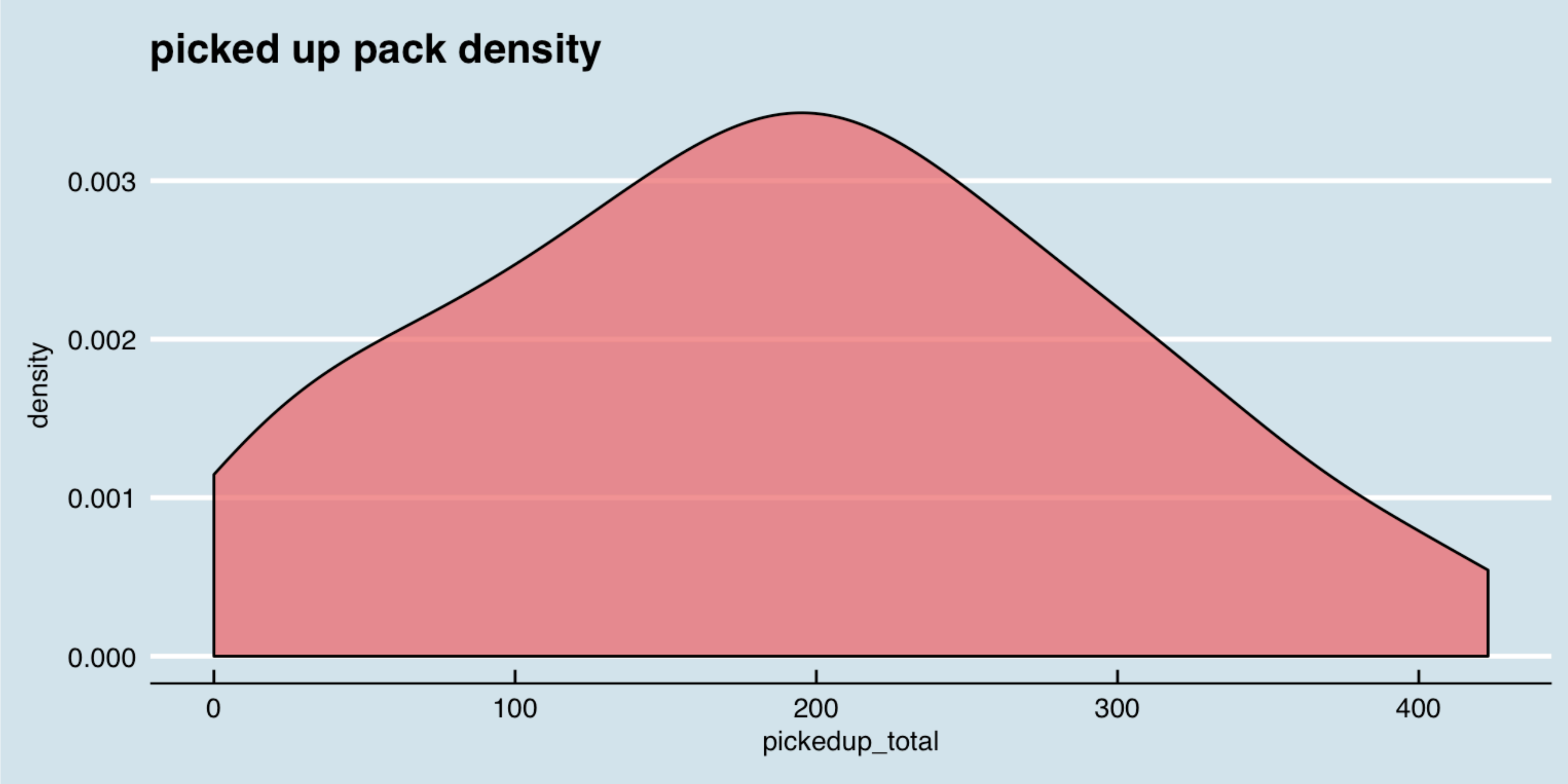


The curve…

TOTAL PICKED UP PACKS PER DRIVER

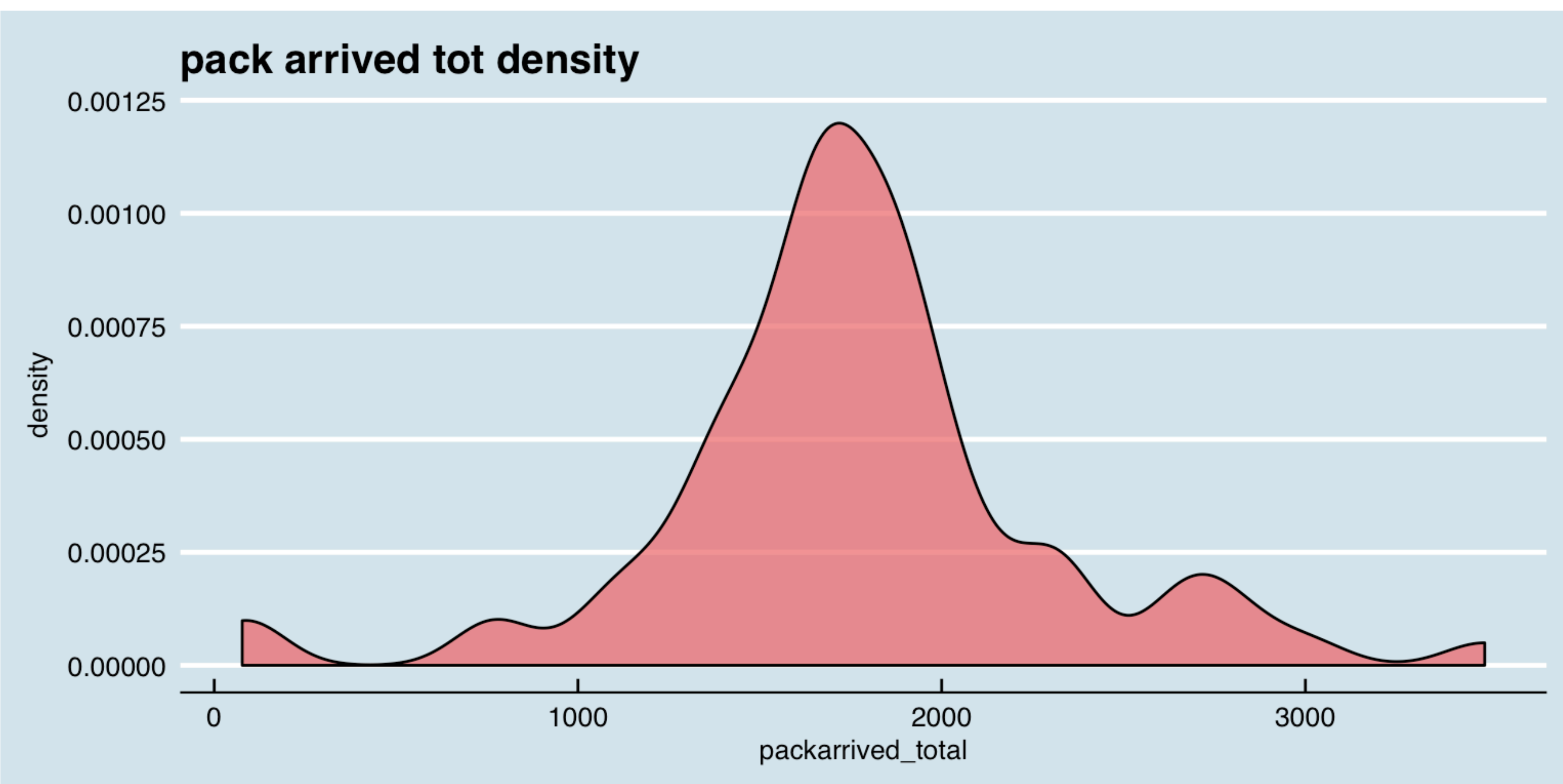
```
aggregate_data6 <- aggregate_data4 %>% group_by(driver_code) %>% summarise(pickedu
p_total = sum(pickup_services))
aggregate_data6 <- aggregate_data6 %>% filter( driver_code != "851", driver_code !
= "805")
```

```
pickedup_plot <- ggplot(data = aggregate_data6, aes(x = pickedup_total ))
pickedup_plot + geom_density(fill = "indianred2", alpha = 0.7) + theme_economist()
+ labs(title = "picked up pack density")
```



TOTAL ARRIVED PACKS PER DRIVER

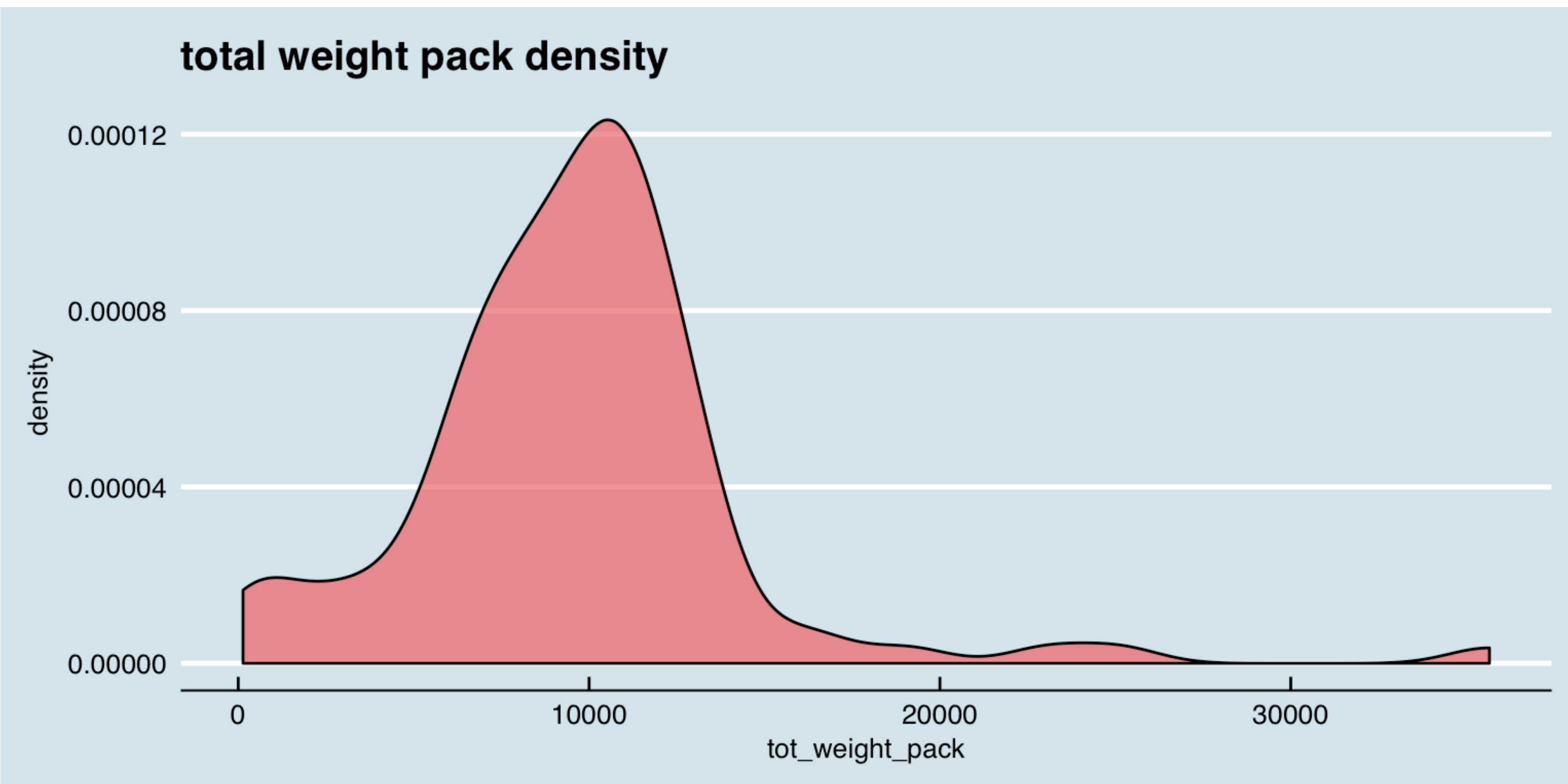
```
aggregate_data7 <- aggregate_data4 %>% group_by(driver_code) %>% summarise(packarr
ived_total = sum(pack_arrived))
setdiff(aggregate_data7$driver_code, aggregate_data2$driver_code)
aggregate_data7 <- aggregate_data7 %>% filter( driver_code != "851", driver_code !
= "805")


packarrived_plot <-ggplot(data = aggregate_data7, aes(x = packarrived_total ))
packarrived_plot + geom_density(fill = "indianred2", alpha = 0.7) + theme_economis
t() + labs(title = "pack arrived tot density")
```



TOTAL WEIGHT OF PACKS DELIVERED BY DRIVER

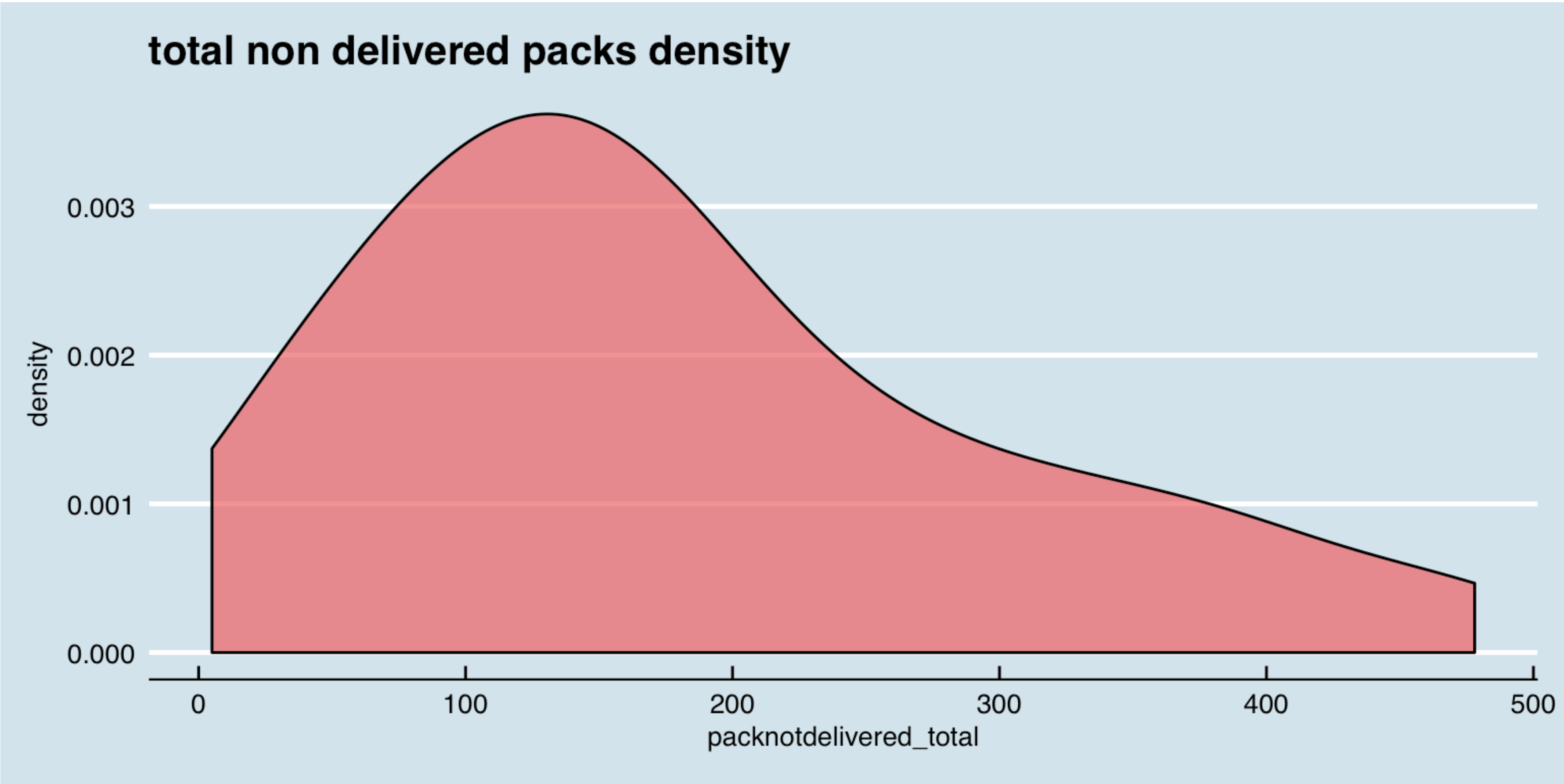
```
data_exploratory <- data_exploratory %>% mutate(weight_pack = gsub(pattern = ",",r
eplacement = ".", x = weight_pack))
data_exploratory$weight_pack <- as.double(data_exploratory$weight_pack)
aggregate_data8 <- data_exploratory %>% group_by(driver_code, day_deliv) %>% summa
rise(sum(weight_pack))
aggregate_data9 <- data_exploratory %>% group_by(driver_code) %>% summarise(tot_we
ight_pack = sum(weight_pack))


summary(aggregate_data9)
```

```
weightpack_plot <-ggplot(data = aggregate_data9, aes(x = tot_weight_pack ))
weightpack_plot + geom_density(fill = "indianred2", alpha = 0.7) + theme_economist
() + labs(title = "total weight pack density")
```



TOTAL PACKS NOT DELIVERED PER DRIVER

```
aggregate_data10 <- aggregate_data4 %>% group_by(driver_code) %>% summarise(packno
tdelivered_total = sum(not_delivered))
setdiff(aggregate_data10$driver_code, aggregate_data2$driver_code)
aggregate_data10 <- aggregate_data10 %>% filter( driver_code != "851", driver_code
!= "805")


nondelivered_plot <-ggplot(data = aggregate_data10, aes(x = packnotdelivered_total
))
nondelivered_plot + geom_density(fill = "indianred2", alpha = 0.7) + theme_economi
st() + labs(title = "total non delivered packs density")
```

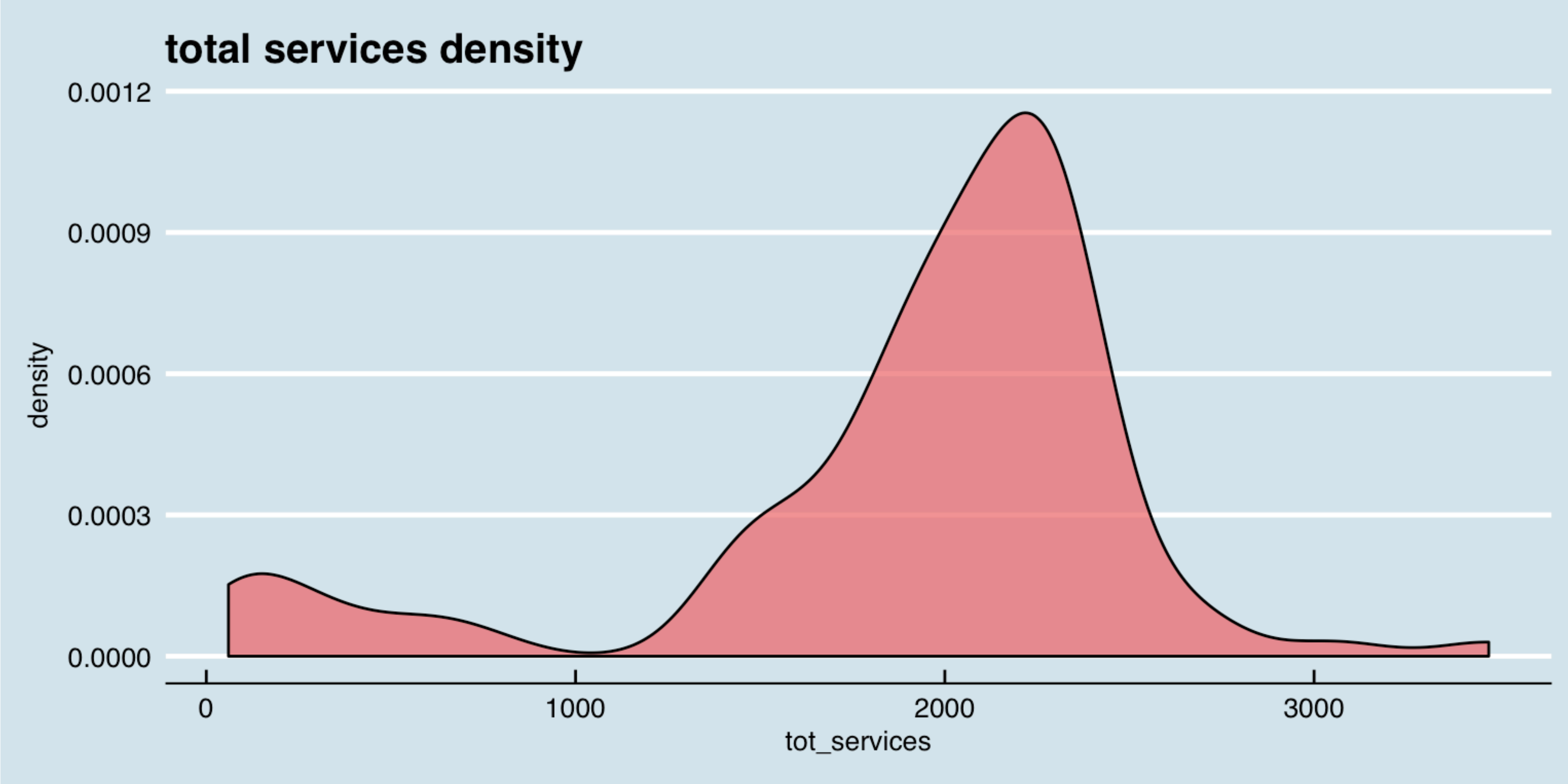


TOTAL SERVICES PER DRIVER

This variable summarise the total amount of services performed by each driver (pack_delivered, pickedup_pack and failed deliveries or pick up).

```
aggregate_data11 <- aggregate_data4 %>% group_by(driver_code) %>% summarise(tot_se
rvices = sum(tot_serv))
setdiff(aggregate_data11$driver_code, aggregate_data2$driver_code)
aggregate_data11 <- aggregate_data11 %>% filter( driver_code != "851", driver_code
!= "805")
summary(aggregate_data11)
```

```
totservices_plot <-ggplot(data = aggregate_data11, aes(x = tot_services ))
totservices_plot + geom_density(fill = "indianred2", alpha = 0.7) + theme_economis
t() + labs(title = "total services density")
```



The lack of precise data regarding the amount of picked up kg represents the reason why this variable will not be taken in consideration. A lot more from this point of view will have to be done in order to obtain the necessary variables to solve this problem.

OBTAINING AN AGGREGATE DATA FRAME WITH ALL THE NECESSARY VALUES

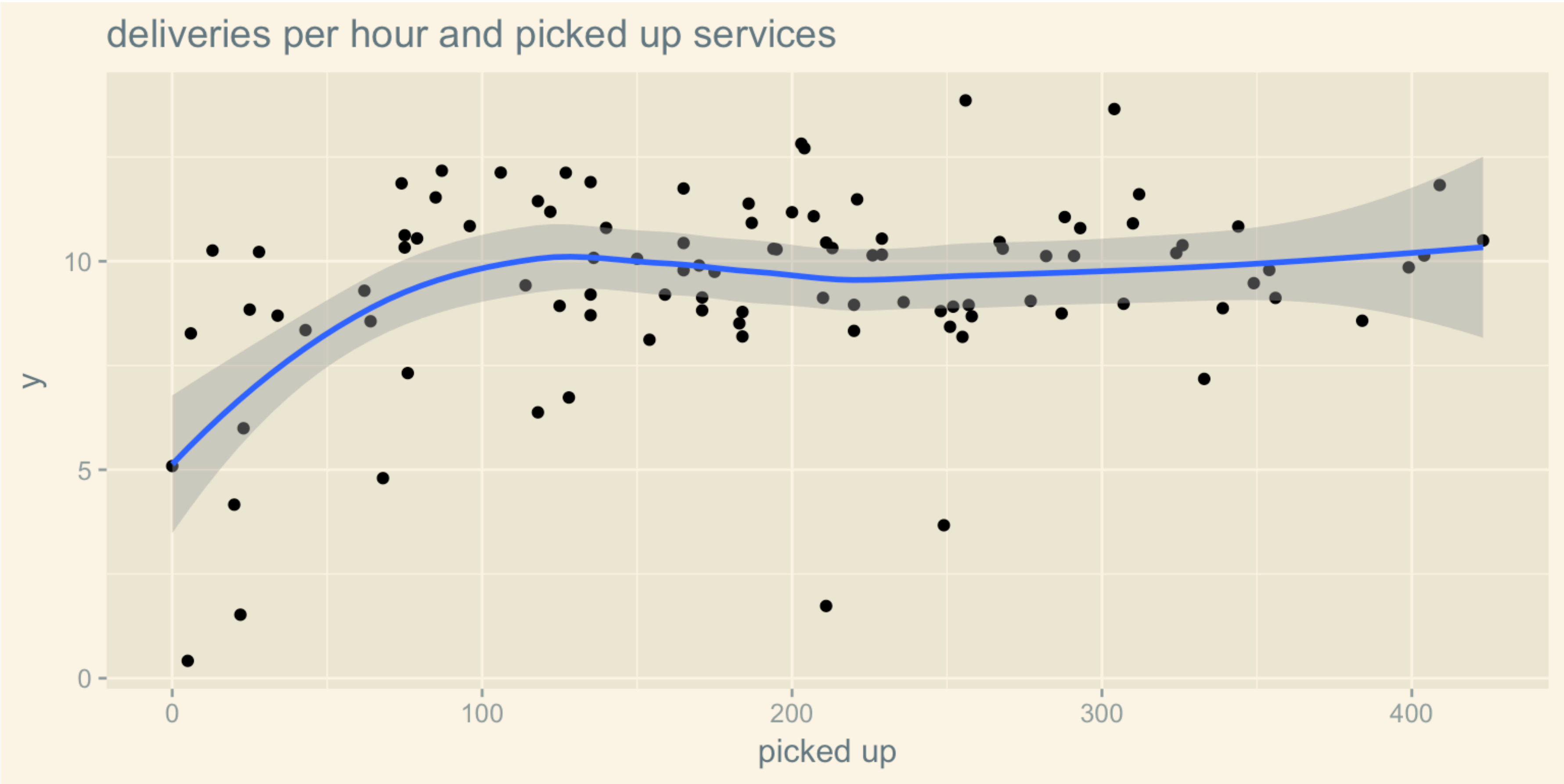
```
aggregate_data_last <- left_join(aggregate_data2,aggregate_data3)
aggregate_data_last <- left_join(aggregate_data_last, aggregate_data3)
aggregate_data_last <- left_join(aggregate_data_last, aggregate_data5)
aggregate_data_last <- left_join(aggregate_data_last, aggregate_data6)
aggregate_data_last <- left_join(aggregate_data_last, aggregate_data7)
aggregate_data_last <- left_join(aggregate_data_last, aggregate_data9)
aggregate_data_last <- cbind(aggregate_data_last, zone_split$del_zone1,zone_split$
del_zone2,zone_split$del_zone3)
aggregate_data_last <- aggregate_data_last %>% rename("del_zone1"
=`zone_split$del_zone1`,
                                                       "del_zone2" = `zone_split$de
l_zone2`,
                                                       "del_zone3" = `zone_split$de
l_zone3`)

aggregate_data_last$`aggregate_data3$day_worked` <- NULL

aggregate_data_last <-cbind(aggregate_data_last, tot_services = aggregate_data11$t
ot_services)
```

ANALYSIS OF THE DEPENDENT VARIABLE COMPARED TO THE

INDEPENTENT VARIABLES

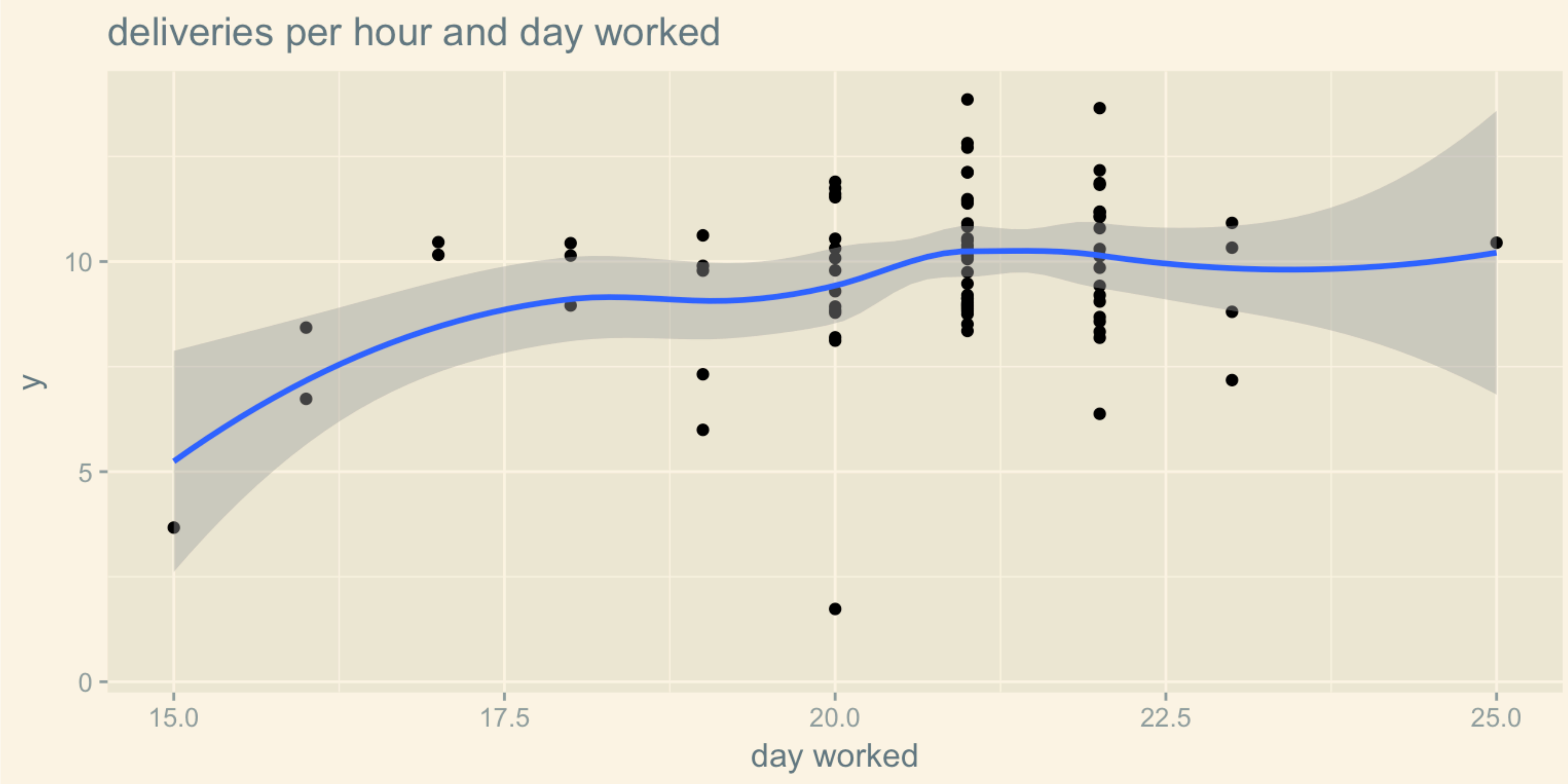
Y AND PICKED UP PACKS

```
y_pickedup_plot<- ggplot(data = aggregate_data_last, aes(x = pickedup_total,y = y)
)+ theme_solarized_2()+
  labs(title = "deliveries per hour and picked up services", x =  "picked up" )

#

y_pickedup_plot + geom_point()+ geom_smooth()
```

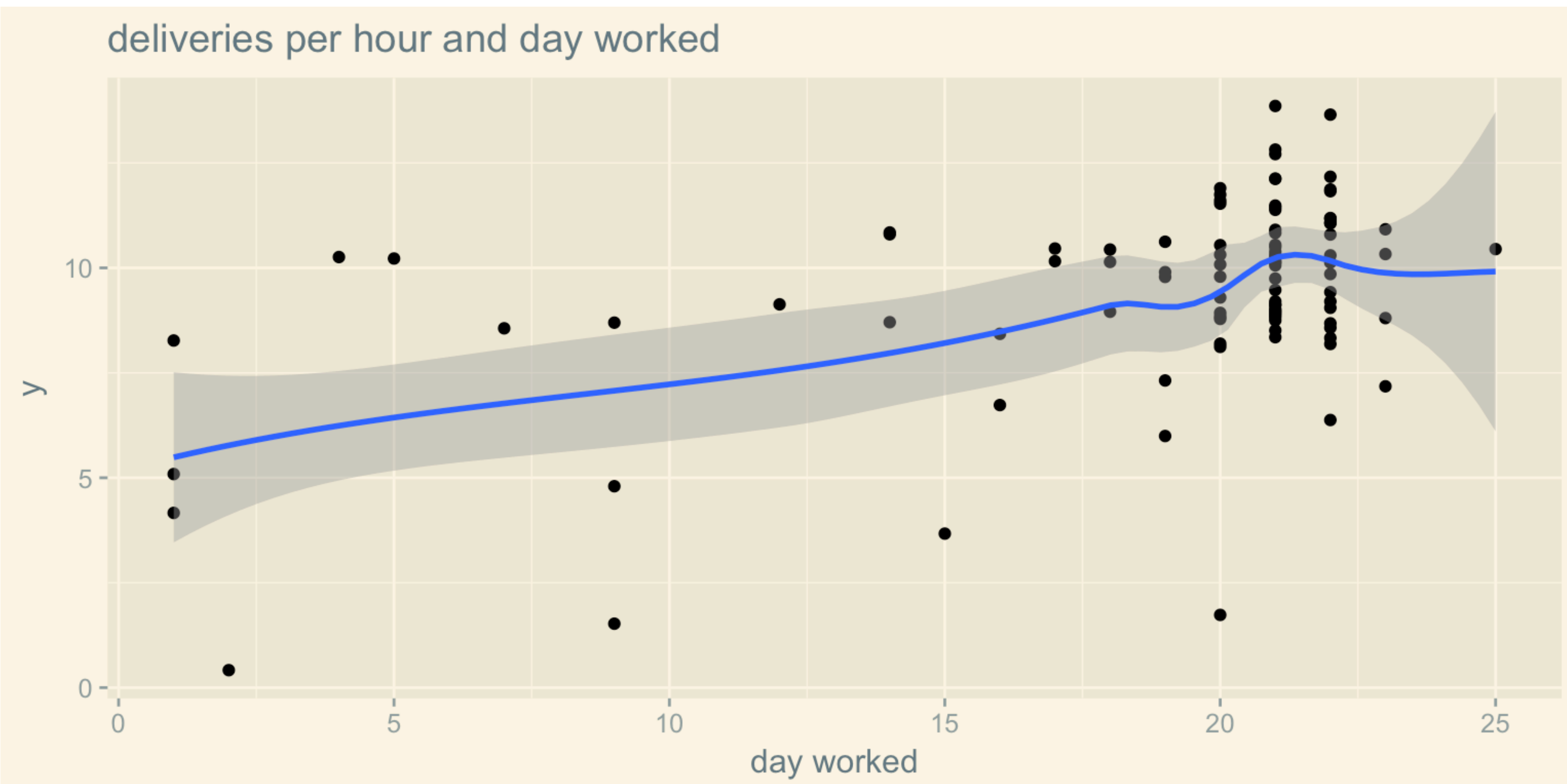


Y AND DAY WORKED

```
y_dayworked_plot<- ggplot(data = aggregate_data_last, aes(x = day_worked,y = y))+
theme_solarized_2()+
  labs(title = "deliveries per hour and day worked", x =  "day worked" )

y_dayworked_plot + geom_point() + geom_smooth() + scale_x_continuous(limits = c(15
,25))
```

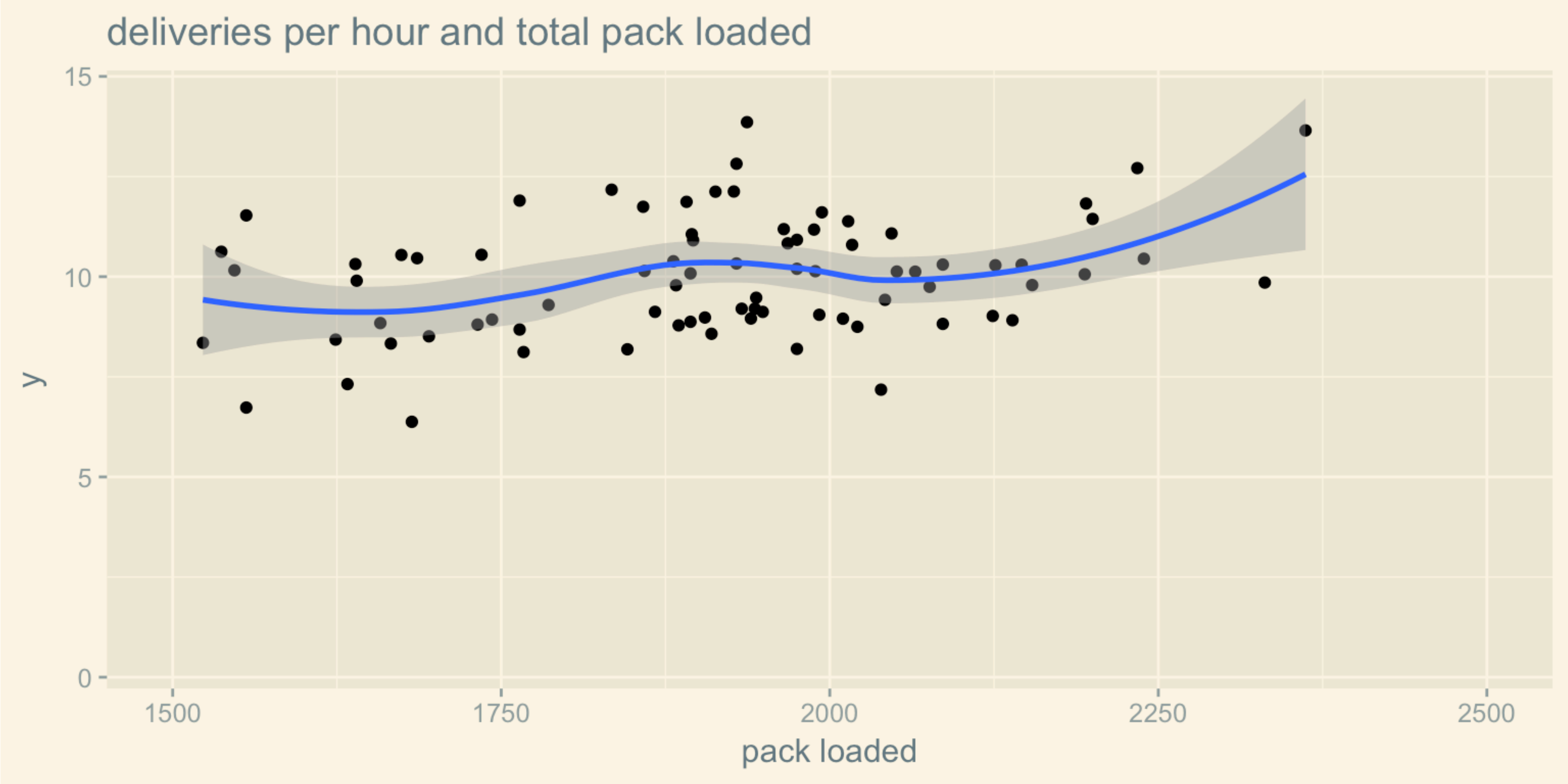



```
y_dayworked_plot + geom_point() + geom_smooth()
```

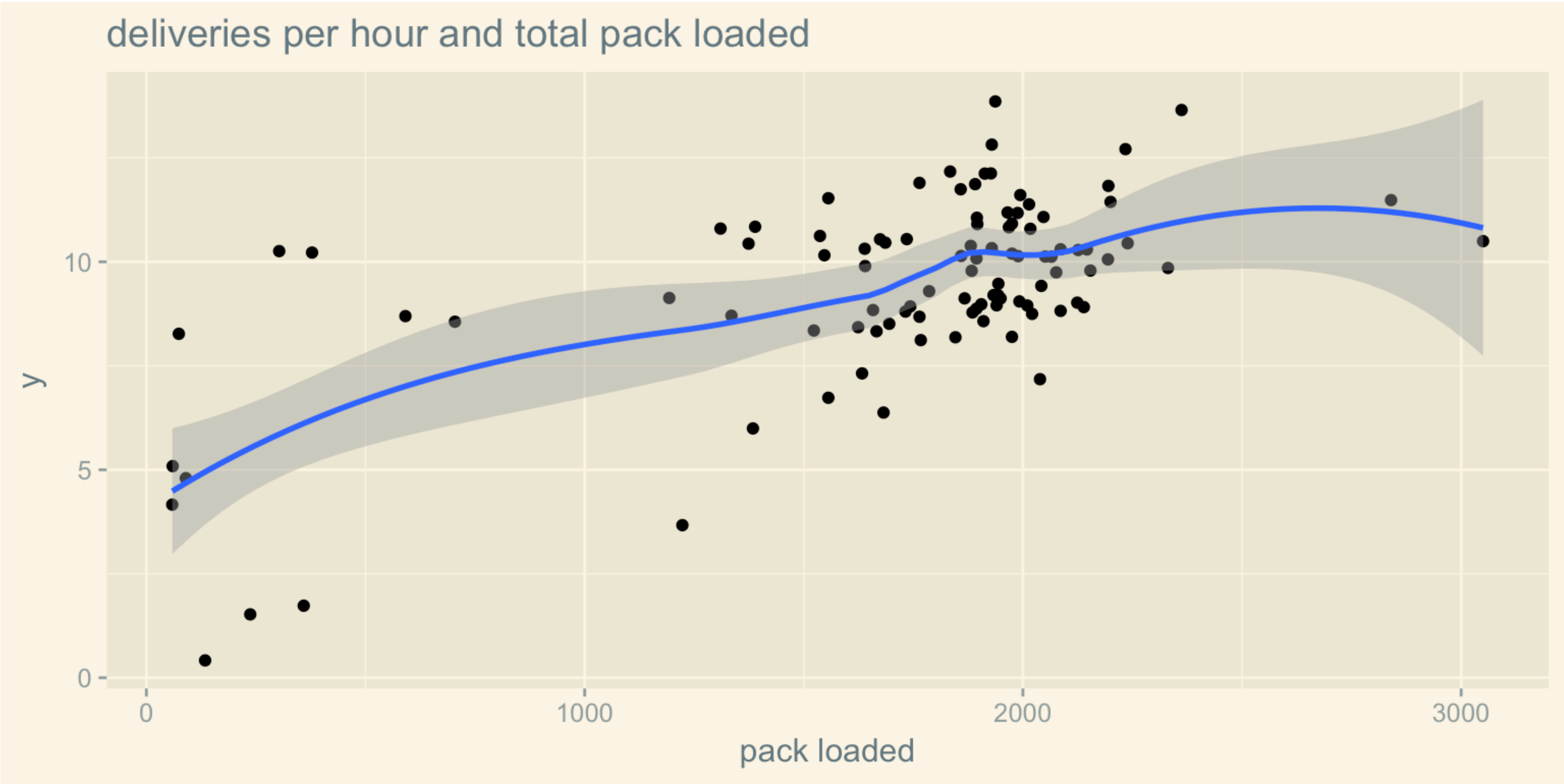


Y AND TOT PACK LOADED

```
y_packloaded_plot<- ggplot(data = aggregate_data_last, aes(x = tot_packloaded,y =
y))+ theme_solarized_2()+
  labs(title = "deliveries per hour and total pack loaded", x =  "pack loaded" )

y_packloaded_plot + geom_point() + geom_smooth()+ scale_x_continuous(limits = c(15
00, 2500))
```

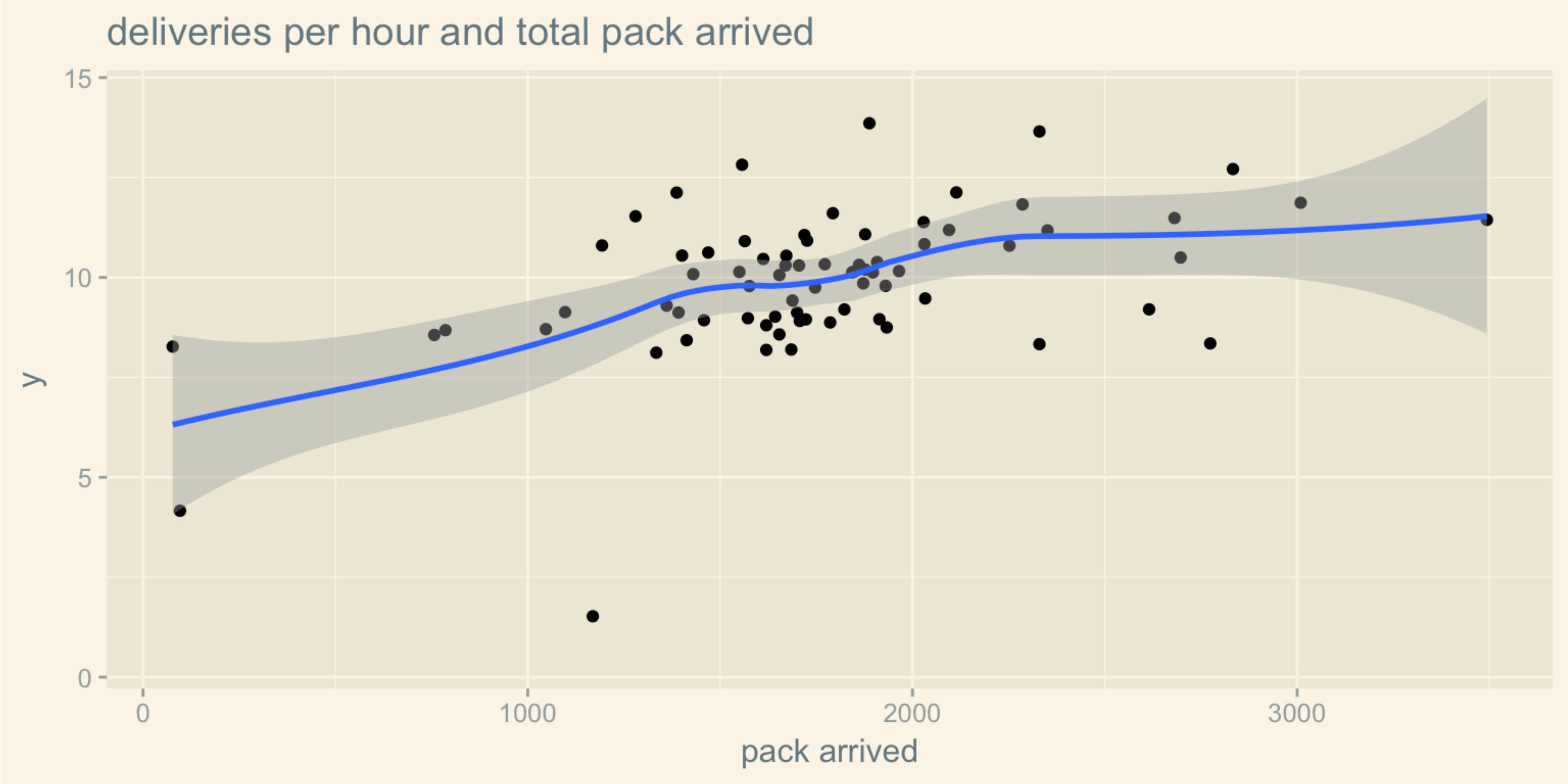



```
y_packloaded_plot + geom_point() + geom_smooth()
```



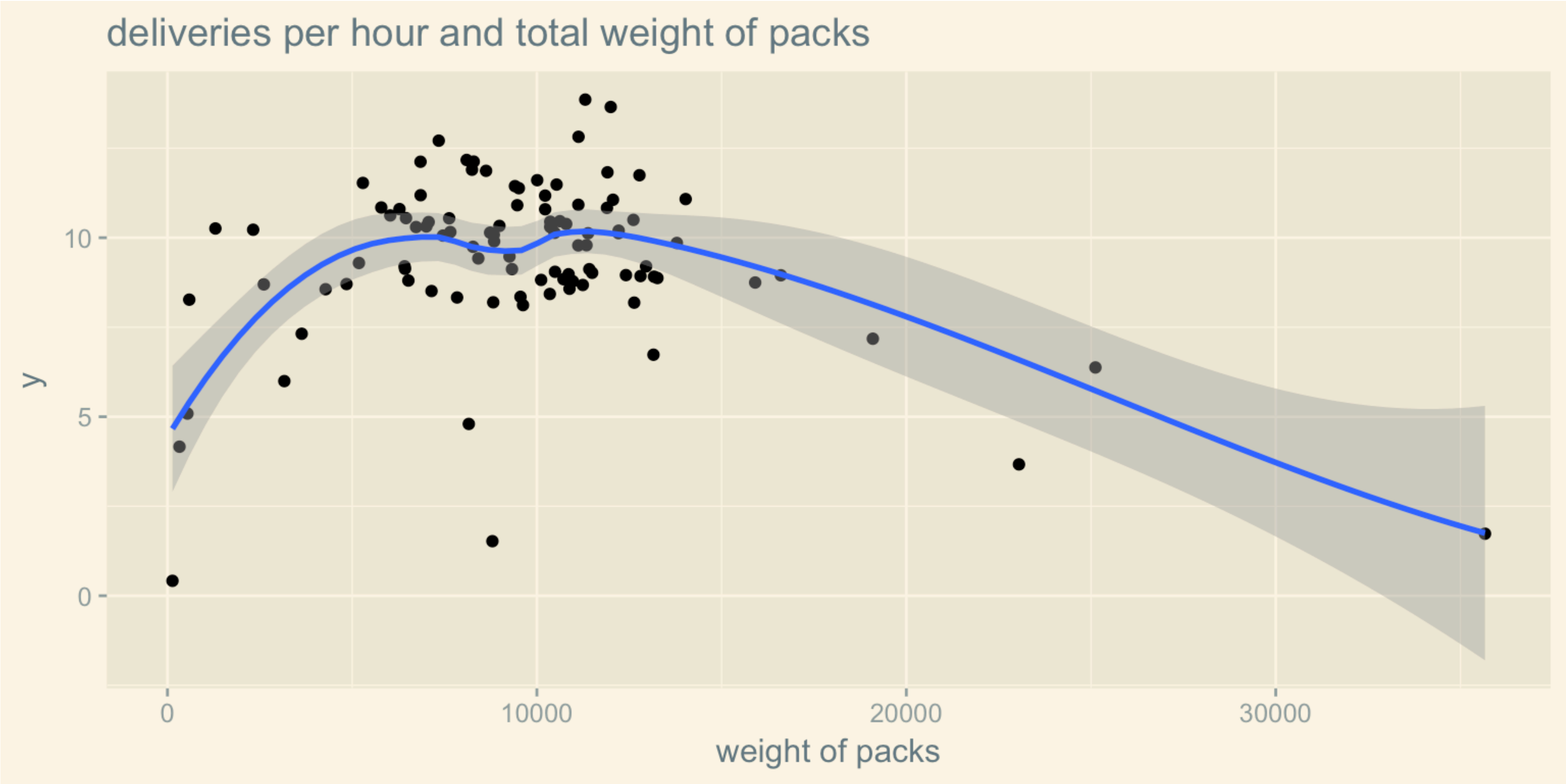
Y AND PACK ARRIVED TOTAL

```
y_packarrived_plot<- ggplot(data = aggregate_data_last, aes(x = packarrived_total, 
y = y))+ theme_solarized_2()+
  labs(title = "deliveries per hour and total pack arrived", x =  "pack arrived" )

y_packarrived_plot + geom_point() + geom_smooth()
```



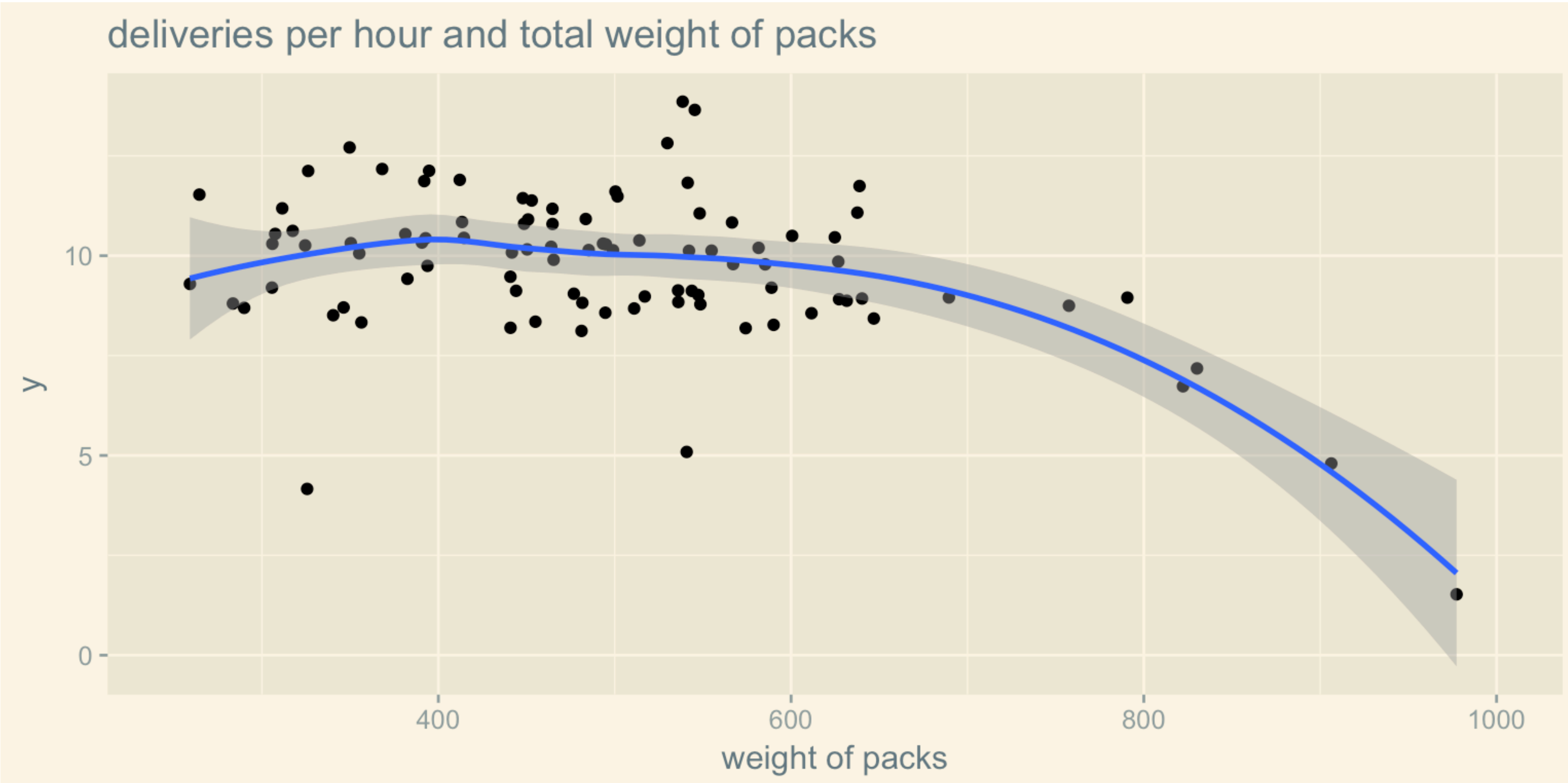
Y AND WEIGHT TOTAL

```
y_weight_plot<- ggplot(data = aggregate_data_last, aes(x = tot_weight_pack,y = y))
+ theme_solarized_2()+
  labs(title = "deliveries per hour and total weight of packs", x =  "weight of pa
cks" )

y_weight_plot + geom_point() + geom_smooth()
```

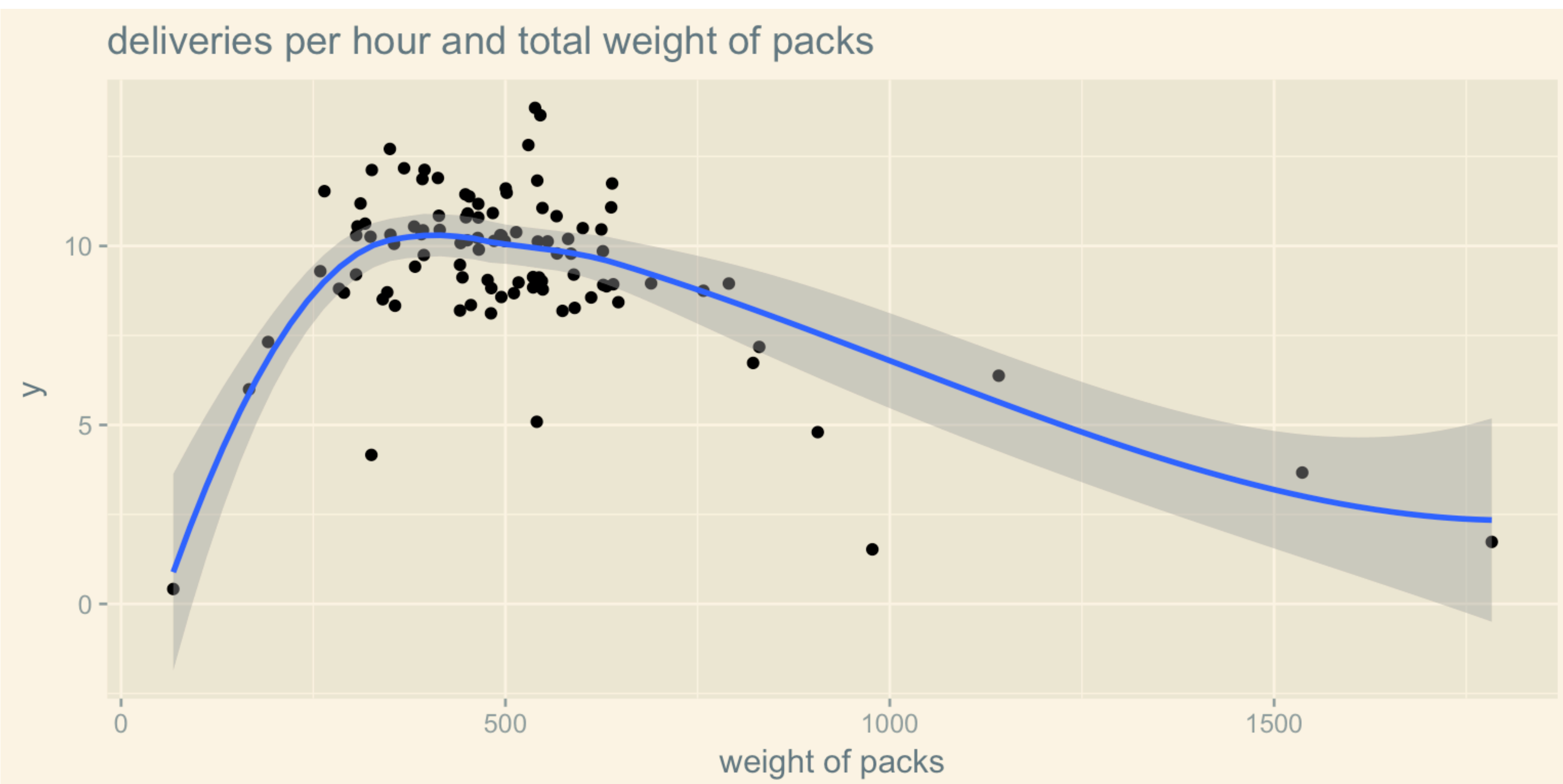


Some driver result in a lower weight due to a lower amount of worked day. As a consequence, I should divide the weight for the number of day worked.

```
y_weight_dayworked_plot<- ggplot(data = aggregate_data_last, aes(x = tot_weight_pa
ck/day_worked,y = y))+ theme_solarized_2()+
  labs(title = "deliveries per hour and total weight of packs", x =  "weight of pa
cks" )

y_weight_dayworked_plot + geom_point() + geom_smooth()+   scale_x_continuous(limit
s = c(250,1000))
```

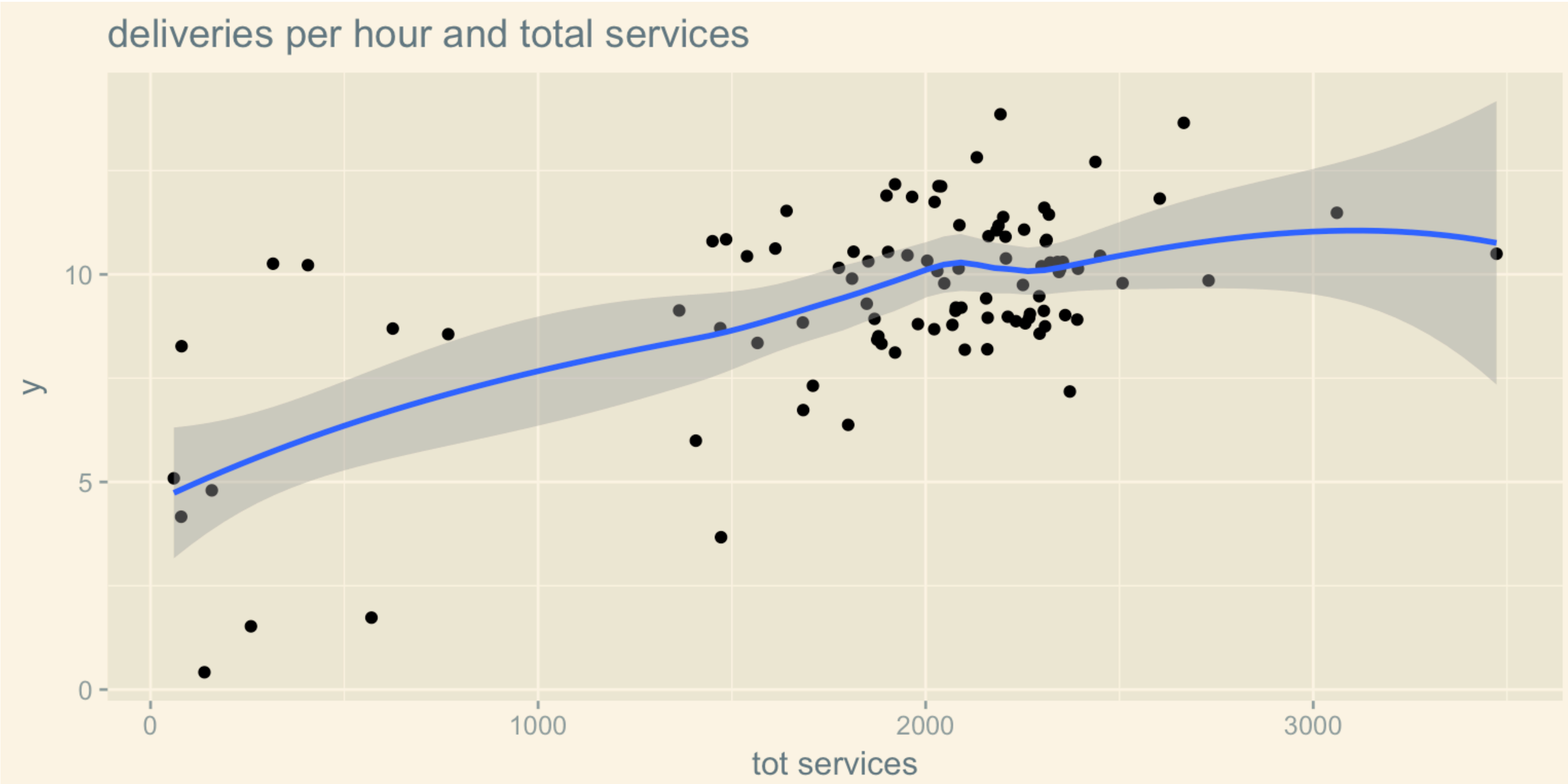



```
y_weight_dayworked_plot + geom_point() + geom_smooth()
```



Y AND TOT SERVICES

```
y_totservices_plot<- ggplot(data = aggregate_data_last, aes(x = tot_services,y = y
))+ theme_solarized_2()+
  labs(title = "deliveries per hour and total services", x =  "tot services" )
y_totservices_plot + geom_point() + geom_smooth()
```



Y AND AREA OF DELIVERY